

НАЦІОНАЛЬНИЙ ТУХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

М.Д. ГЕРАЇМЧУК, Ю.Ф. ЛАЗАРЄВ , Т.О. ТОЛОЧКО

МОДЕЛЮВАННЯ СИСТЕМ У СЕРЕДОВИЩІ MATLAB-SIMULINK

КОМП'ЮТЕРНИЙ ПРАКТИКУМ

ЗАТВЕРДЖЕНО

На засіданні Вченої Ради ПБФ

Протокол №14\05 від 26.12.2005р.

КИЇВ 2006

М.Д. Гераїмчук, Ю.Ф. Лазарєв, Т.О. Толочко Моделювання систем у середовищі MATLAB-SIMULINK: Комп'ютерний практикум. – К.:, 2006. – 175с.

Укладачі: М.Д. Гераїмчук, д.т.н., проф.,
Ю.Ф. Лазарєв, к.т.н., доцент,
Т.О. Толочко

Рецензент Глоба

Відповідальний редактор Нікітін О.К., к.т.н., доцент

ЗМІСТ

1. Головні відомості про систему MATLAB і SIMULINK.....	6
1.1. Знайомство з системою MATLAB.....	6
1.2. Пакет візуального програмування SIMULINK.....	11
1.2.1. Бібліотека блоків SIMULINK.....	12
1.2.2. Створення блок-схеми S-моделі.....	23
1.2.3. Встановлення параметрів блоків.....	25
1.2.4. Встановлення параметрів моделювання.....	27
1.2.5. Проведення моделювання і одержання результатів.....	32
Лабораторна робота №1. Технологія моделювання у середовищі SIMULINK.....	36
Лабораторна робота № 2. Формування та дослідження стандартних вхідних інформаційних сигналів, що використовуються при визначенні динамічних характеристик приладів.....	41
Лабораторна робота №3. Моделювання механічних пристроїв з суттєвими нелінійностями	66
Лабораторна робота №4. Моделювання із застосуванням операцій диференціювання та інтегрування.....	75
Лабораторна робота №5. Моделювання роботи логічних схем	84
Лабораторна робота № 6. Моделювання та аналіз випадкових збурень, що впливають на роботу приладів.....	91
Лабораторна робота №7. Динамічний аналіз систем автоматичного керування.....	96
Перелік блоків SIMULINK.....	127
Блоки SIMULINK, розсортовані за алфавітом.....	165
Список літератури	174

Навчальний посібник призначений для студентів вищих технічних навчальних закладів. Він присвячений викладенню основ практичного засвоєння засобів математичної системи MATLAB і пакета Simulink, який входить до складу цієї системи, у частині, що торкається створення математичних обчислювальних моделей і проведення моделювання так званих динамічних систем.

Динамічні системи складають основу більшості створюваних технічних систем (машин, механізмів, приладів, систем автоматичного керування тощо). Під ними зазвичай розуміють такі реальні фізичні об'єкти, поведінка яких задовільно можна описати за допомогою системи звичайних диференціальних рівнянь (ЗДР), аргументом яких є час.

Головною задачею проектування технічної системи є передбачення її поведінки у часі у заданих умовах (аналіз) і таке змінення її конструктивних параметрів, яке б забезпечувало задані характеристики її поведінки у цих умовах (синтез). Аналіз складної технічної системи найчастіше зводиться або до теоретичного дослідження розв'язків ЗДР, що описують її поведінку, або до чисельного математичного моделювання системи, тобто моделювання процесів, що протікають у ній. Саме питанням утворення математичних обчислювальних моделей динамічних систем і проведення моделювання по них і присвячений цей практикум.

У наш час, у зв'язку з появою і доступністю користувачам персональних комп'ютерів потужних математичних систем, які увібрали у себе новітні досягнення інженерної думки, з'явилися й нові можливості проведення аналізу і синтезу технічних систем. Серед нових обчислювальних засобів аналізу технічних систем однією з найбільш пристосованих для розв'язування інженерних задач є математична комп'ютерна система MATLAB. В її склад входить багато пакетів

прикладних програм, які значно розширюють коло розв'язуваних системою задач і наближають її до інженерної практики. Зокрема, до таких пакетів можна віднести пакет Control Toolbox, який значно спрощує аналіз і синтез систем автоматичного керування, що описуються лінійними диференціальними рівняннями з постійними коефіцієнтами; пакет Signal Processing, який дозволяє проводити цифрову обробку даних, проектувати фільтри сигналів і моделювати поведінку цих фільтрів в різноманітних умовах.

Особливо зручним для проведення моделювання процесів в динамічних системах є пакет візуального програмування Simulink, який дозволяє моделювати процеси у складних (у тому числі нелінійних) неперервних і дискретних динамічних системах, обходячи процес написання програм, шляхом складання у середовищі Simulink блок-схеми системи з окремих готових візуально оформлених блоків і з'єднанню їх між собою у визначеному порядку. Така процедура одержала назву візуального програмування. Візуальне програмування суттєво спрощує процес створення обчислювальної програми, не потребує знань мов програмування, дозволяє наочно подати структуру моделюваної системи і робить процес моделювання вельми наочним і легко корегованим.

1. ГОЛОВНІ ВІДОМОСТІ ПРО СИСТЕМУ MATLAB I SIMULINK

Пакет Simulink працює у середовищі математичної системи MATLAB. Тому спочатку наведемо деякі головні відомості про цю систему.

1.2. ЗНАЙОМСТВО З СИСТЕМОЮ MATLAB

Після виклику системи MATLAB версії 6.5 на екрані виникає вікно, подане на рис. 1.

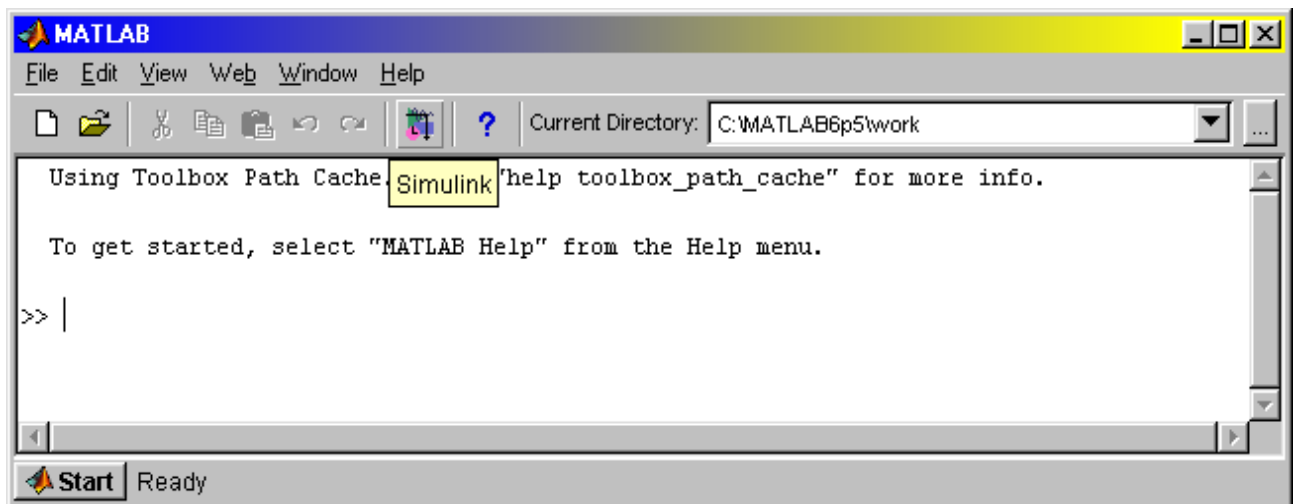


Рис.1. Командне вікно MATLAB

У цьому вікні у подальшому вводяться команди, оператори і дані для подальших обчислень. В нього ж виводяться результати здійснених обчислень. Ознакою того, що система готова для обчислень є наявність у останньому рядку вікна знака запрошення ">>" і міготливої вертикальної риси поряд з ним.

Обчислення у командному вікні можуть здійснюватися шляхом введення після знака запрошення відповідного арифметичного виразу без знака присвоювання (у MATLAB знаком присвоювання є "="), а також шляхом запису у командний рядок команди або оператора зі знаком присвоювання. Приклади наведені на рис. 2.

Значення усіх введених і обчислених змінних зберігаються до кінця сеансу роботи з системою у так званому робочому просторі системи – Workspace.

Особливістю MATLAB є те, що усі числові дані сприймаються системою як матриці і дії між ними здійснюються як дії над матрицями. При цьому окреме число розглядається як матриця розміром 1×1 , а послідовність N чисел (вектор) – як матриця-рядок розміром $1 \times N$.

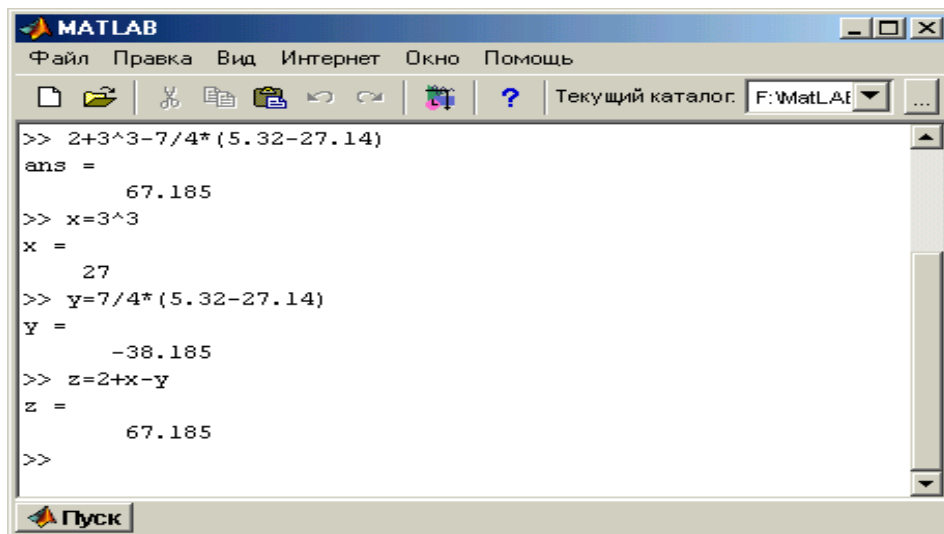


Рис.2. Приклади виконання обчислень у командному вікні MATLAB

Введення числового вектора здійснюється поелементно, у квадратних дужках (рис. 3). Окремі елементи розділяються між собою пропусками, або комами.

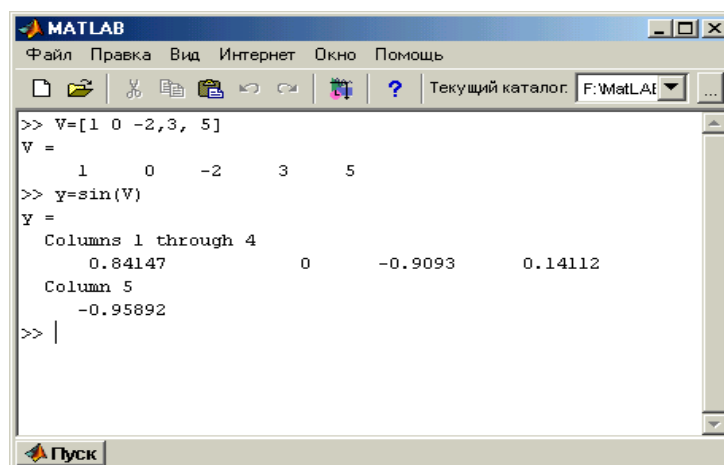


Рис. 3. Введення вектора і обчислення функції від вектора

Усі вбудовані алгебричні функції, якщо їхнім аргументом є вектор або матриця, обчислюють вектор або матрицю тих самих розмірів, що й аргумент, кожний елемент яких є значенням відповідної функції при значенні відповідного елемента аргумента (див. рис. 3). Введення значень елементів матриці також здійснюється у квадратних дужках порядково. Рядки відділяються один від одного знаком ";" (рис. 4).

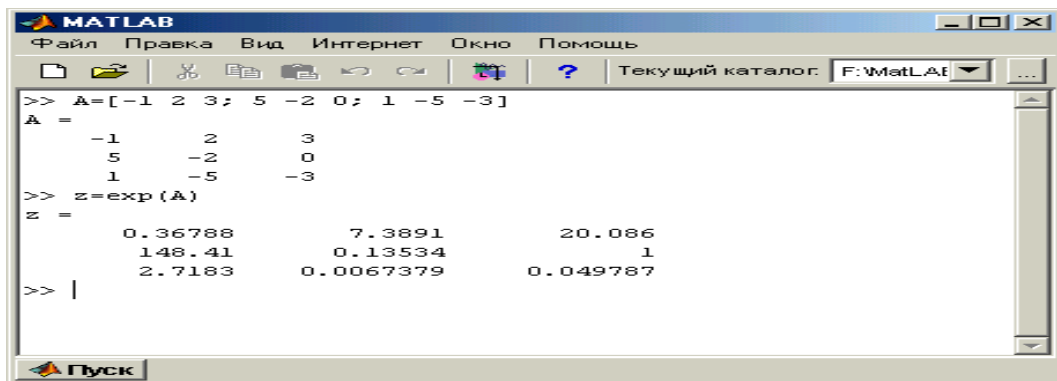


Рис. 4. Введення матриці і обчислення функції від матриці

Окрім того усі числові елементи матриць можуть бути комплексними. При цьому для позначення уявної одиниці можна використовувати символи "i" та "j". Усі вбудовані алгебричні функції MATLAB здійснюють обчислення при комплексних значеннях аргумента і результат у цьому випадку є також комплексним (рис. 5).

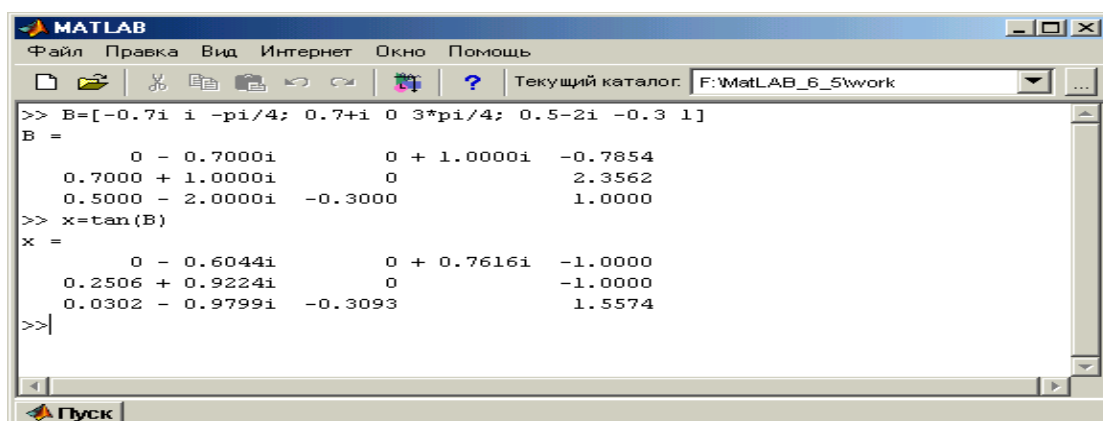


Рис. 5. Введення значень і обчислення функції від комплексного матричного аргумента

Вельми зручною у MATLAB є побудова графіків функцій. Наприклад, щоб побудувати графік функції $y=e^{-hx}\sin(x)$ у діапазоні змінювання аргумента $0 \leq x \leq 6\pi$ при $h=0,2$ достатньо ввести у командне вікно таку послідовність операторів:

```
>> h=0.2;  
>> x=0:0.02:6*pi;  
>> y=exp(-h*x).*sin(x);  
>> plot(x,y), grid  
>> title('Графік функції y=exp(-0.2*x)*sin(x)')  
>> xlabel('x'),ylabel('y')
```

В результаті на екрані виникне нове вікно з зображенням графіка (рис. 6). Його називають фігурою.

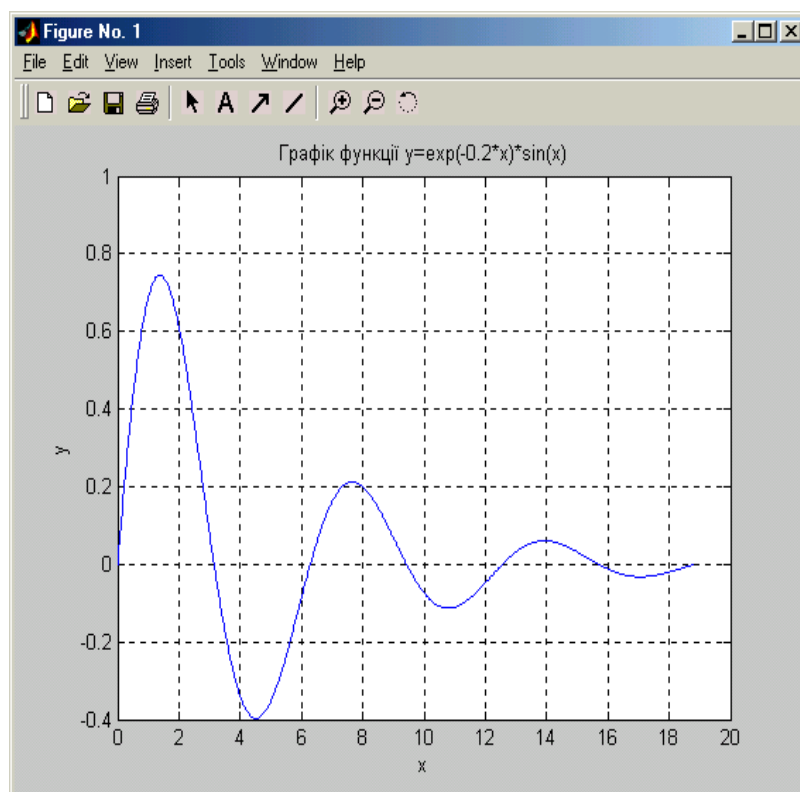


Рис. 6. Виведення графіка функції у вікно фігури

Завдяки вказаним особливостям і наявності великої кількості вбудованих функцій (процедур) система MATLAB навіть при роботі з нею у командному режимі є одним з найпотужніших наукових калькуляторів. У середовищі MATLAB можна створювати і програми, тексти яких записуються мовою програмування високого рівня, яку називають М-мовою. М-мова значною мірою нагадує мову BASIC, перш за все простотою і відсутністю необхідності спеціально оголошувати типи використовуваних змінних. Але, на відміну від BASIC, М-мова має об'єктно-орієнтовану спрямованість, як і мова Pascal, що дозволяє користувачеві утворювати власні класи обчислювальних об'єктів, будувати для них власні процедури дій над об'єктами цих класів, користуючись при цьому звичними знаками арифметичних і логічних дій. Програми MATLAB записуються за допомогою вбудованого (або стороннього) текстового редактора у файл, який називають М-файлом. Виклик до виконання програми здійснюють шляхом запису ймення відповідного М-файла у командному рядку командного вікна MATLAB.

Використання програм замість командного режиму має суттєві переваги:

- завдяки тому, що послідовність операторів зберігається на диску, можливо неодноразово звертатися до їхнього виконання, що суттєво спрощує і скорочує процес відлагоджування програми;
- у програмі можна забезпечити організацію процесу введення і змінювання первісних даних у діалоговому режимі; для цього передбачені кілька спеціальних команд і функцій;
- програми можна використовувати неодноразово, отже проводити дослідження при різних вхідних даних;

- окремі, найбільш вдалі і типові частини програми можна використовувати при складанні інших програм, що значно полегшує складання нових програм.

1.2. ПАКЕТ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ SIMULINK

Пакет Simulink дозволяє здійснювати досліджування (моделювання у часі) поведінки динамічних лінійних і нелінійних, неперервних і дискретних систем, причому складання «програми» і введення характеристик досліджуваних систем здійснювати у діалоговому режимі, шляхом графічного складання на екрані схеми з'єднань елементарних (стандартних або створених користувачем) блоків. В результаті такого складання виходить модель досліджуваної системи, яку у подальшому називатимемо S-моделлю і яка зберігається у файлі з розширенням **.mdl**. Такий процес утворення обчислювальних програм прийнято називати візуальним програмуванням.

Виклик пакету Simulink можна здійснювати двома способами:

- використовуючи кнопку на панелі інструментів командного вікна MATLAB (див. рис. 1);
- шляхом введення у командний рядок команди "simulink".

В обох випадках на екрані має виникнути вікно браузера бібліотек Simulink, наведене на рис. 7.

Як бачимо у браузері зберігається велика кількість бібліотек. З них найважливішою є бібліотека з тим самим іменем, що й сам пакет, - бібліотека SIMULINK. Вона є ядром пакета, і без її використання не можна створити жодну S-модель.

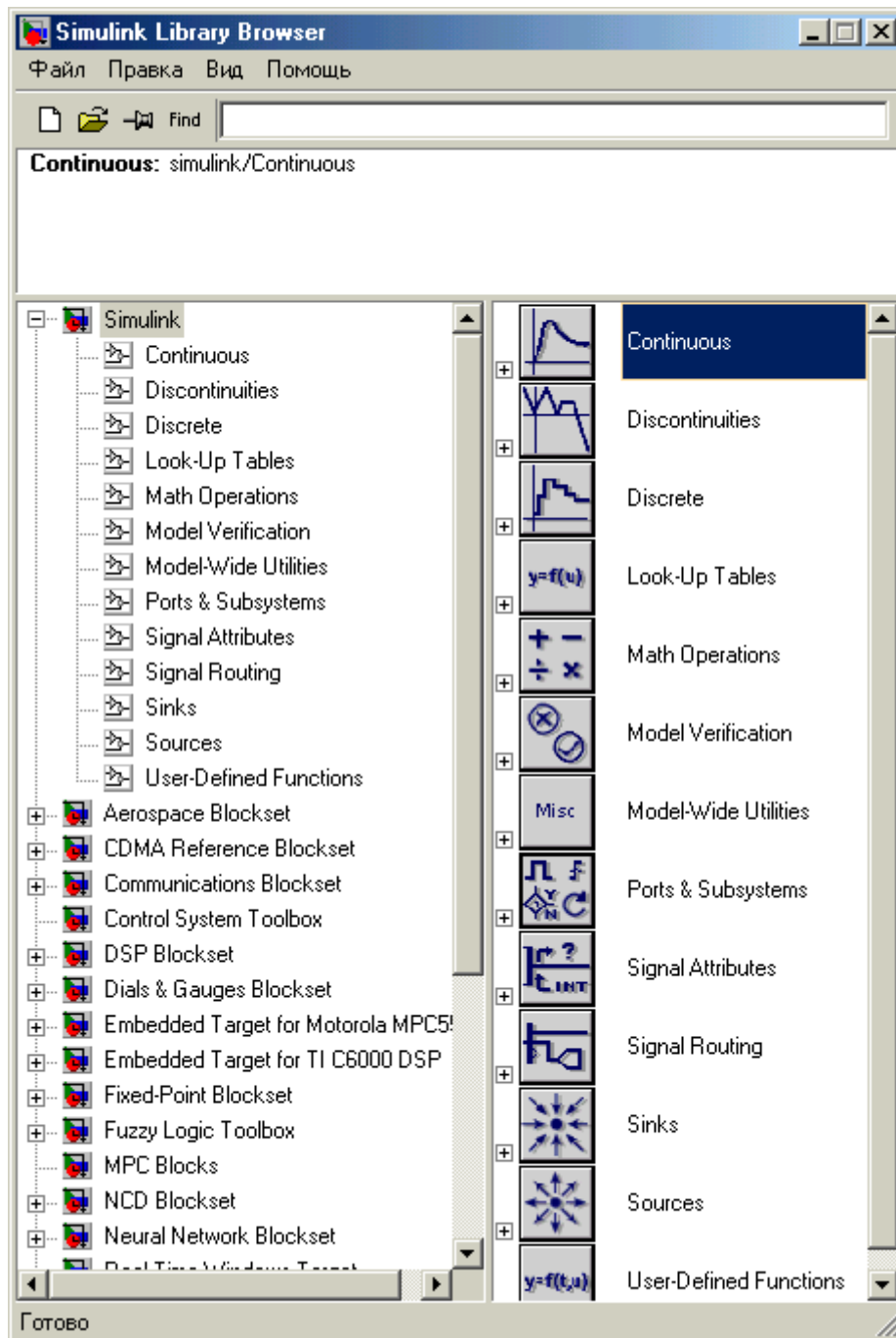


Рис. 7. Вікно браузера бібліотек Simulink

1.2.1. БІБЛІОТЕКА БЛОКІВ SIMULINK

Щоб ознайомитися з розділами бібліотеки SIMULINK, слід скористатися контекстним меню, натиснувши праву кнопку миші при наведенні її курсора на ім'я бібліотеки у браузері бібліотек, а після появи

кнопки меню натиснути ліву кнопку миші на кнопці меню. При цьому виникне вікно, показане на рис. 8.

У цьому вікні наведені піктограми окремих розділів бібліотеки SIMULINK. Подвійно клацаючи мишкою на відповідній піктограмі, можна викликати появу на екрані вікна з зображеннями блоків, що входять у цей розділ.

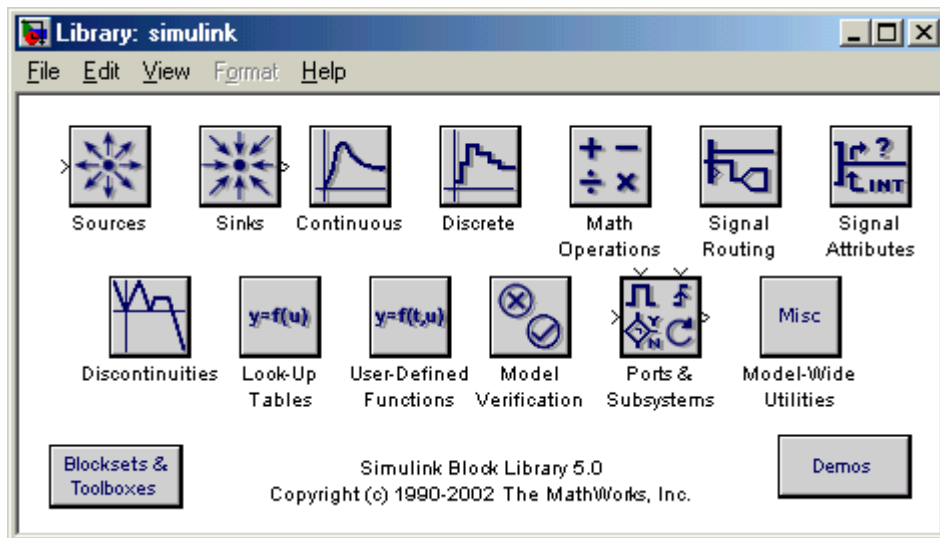


Рис. 8. Вікно бібліотеки SIMULINK

Всього, як впливає з рис.8, бібліотека SIMULINK складається з 15 розділів. Тринадцять з них є головними і не можуть змінюватися користувачем:

Sources	Джерела
Sinks	Приймачі
Continuous	Неперервні елементи
Discrete	Дискретні елементи
Math Operations	Математичні операції
Signals Routing	Пересилання сигналів
Signals Attributes	Атрибути сигналів
Discontinuities	Розривні елементи
Look Up Tables	Табличні функції

User Defined Functions	Функції, що визначаються користувачем
Model Verification	Перевірка моделей
Ports & Subsystems	Порти і підсистеми
ModelWide Utilities	Утилити розширювання моделі
Розділ Sources (Джерела) містить блоки, які генерують вихідну величину блока як деяку певну залежність від модельного часу (рис. 9).	

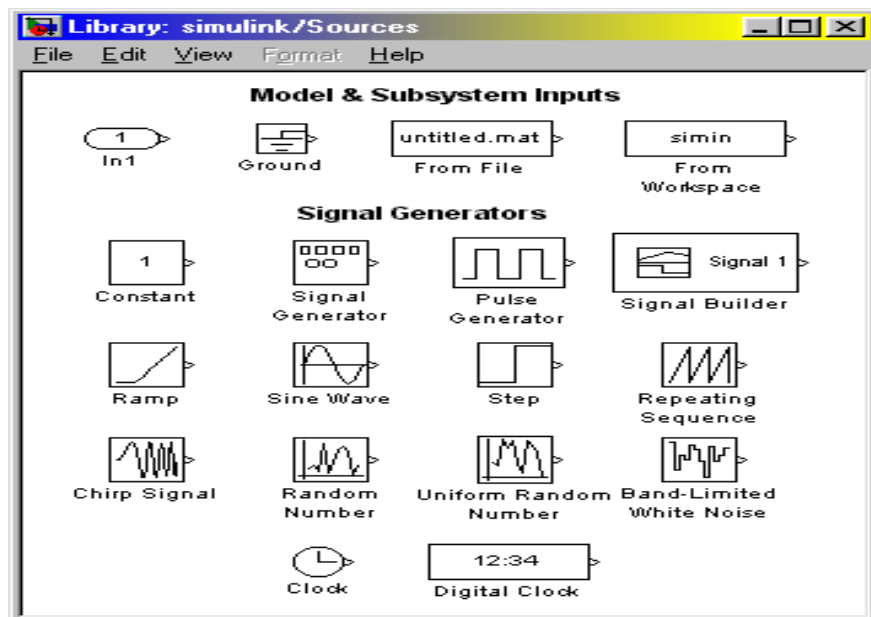


Рис. 9. Вікно розділу Sources бібліотеки SIMULINK

Особливістю блоків цього розділу є те, що вони не мають входів (лише виходи).

Вміст розділу sinks (Приймачі) показаний на рис. 10.

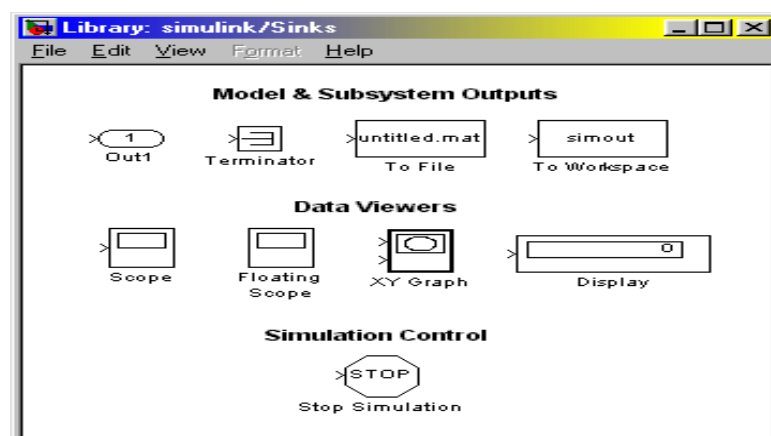


Рис. 10. Вікно розділу Sinks бібліотеки SIMULINK

У цей розділ вміщені блоки, які тільки сприймають сигнали, тобто мають тільки вхід і не мають виходу. Блоки призначені задля зберігання значень вхідного сигналу або подання у графічній формі його залежності від модельного часу чи одного сигналу від іншого. Умовно їх можна поділити на три вида:

- блоки, що використовуються як оглядові вікна при моделюванні;
- блоки, що забезпечують зберігання проміжних і кінцевих результатів моделювання;
- блок керування моделюванням, який дозволяє переривати моделювання при виконанні тих чи інших умов.

Блоки усіх інших розділів, в основному являють собою перетворювачі величини, що подається на вхід блоку у величину, яка одержується на виході блока.

Так, розділ **Continuous** (Неперервні елементи) містить блоки, які забезпечують неперервне перетворення вхідного сигналу у вихідний (рис.11).

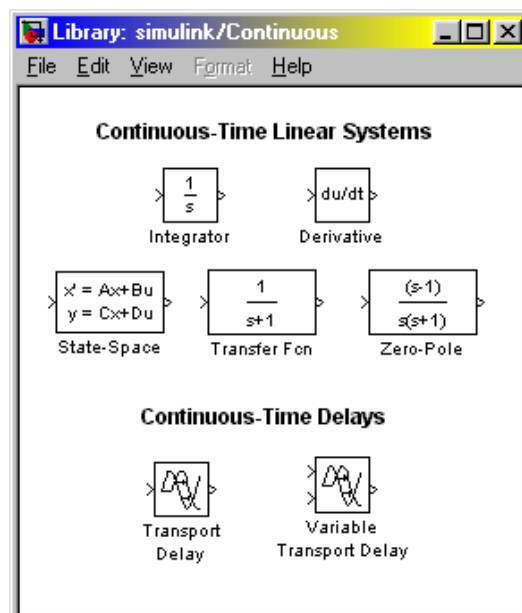


Рис. 11. Вікно розділу Continuous бібліотеки SIMULINK

Блоки цього розділу можна умовно поділити на три групи:

- блоки загального призначення (інтегратори, диференціатори);
- блоки затримки сигналу;
- блоки лінійних стаціонарних ланок.

На відміну від нього розділ **Discrete** складають блоки (рис. 12), які здійснюють перетворення дискретного у часі сигнала, причому лише в окремі моменти часу, які відділені один від одного на деяку певну сталу величину, що називається *дискретом часу* (Sample time).

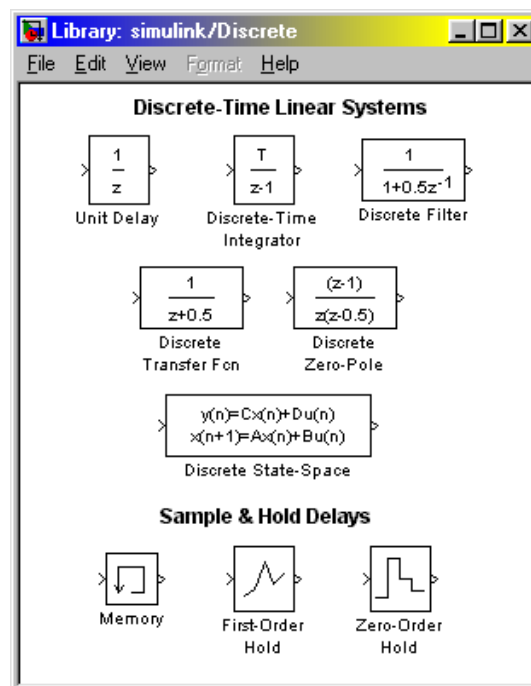


Рис. 12. Вікно розділу Discrete бібліотеки SIMULINK

На рис. 13 поданий вміст розділу **Math Operations** бібліотеки SIMULINK. У цьому розділі зосереджені блоки, які забезпечують типові алгебричні або логічні перетворення вхідних величин у вихідні.

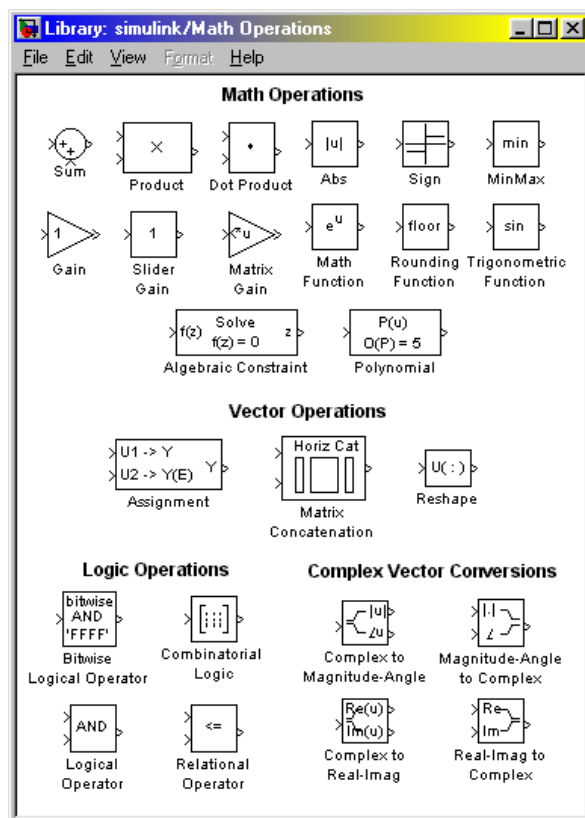


Рис. 13. Вікно розділу Math Operations бібліотеки SIMULINK

Двадцять блоків цього розділу складаються з кількох груп:

- блоки, що реалізують елементарні математичні операції (множення, підсумовування різних математичних об'єктів);
- блоки, що реалізують елементарні математичні функції;
- блоки, які забезпечують логичну обробку вхідних сигналів;
- блоки, які перетворюють комплекснозначний сигнал у два дійсних і навпаки тим чи іншим способом;
- блок, що реалізує відшукування нуля алгебричної функції.

У розділ **Signals Routing** (рис. 14) включені блоки, що забезпечують різного роду пересилання сигналів, таких як переключення сигналів, об'єднання кількох сигналів в єдиний векторний сигнал (шину), розведення сигналів з шини и т. п.

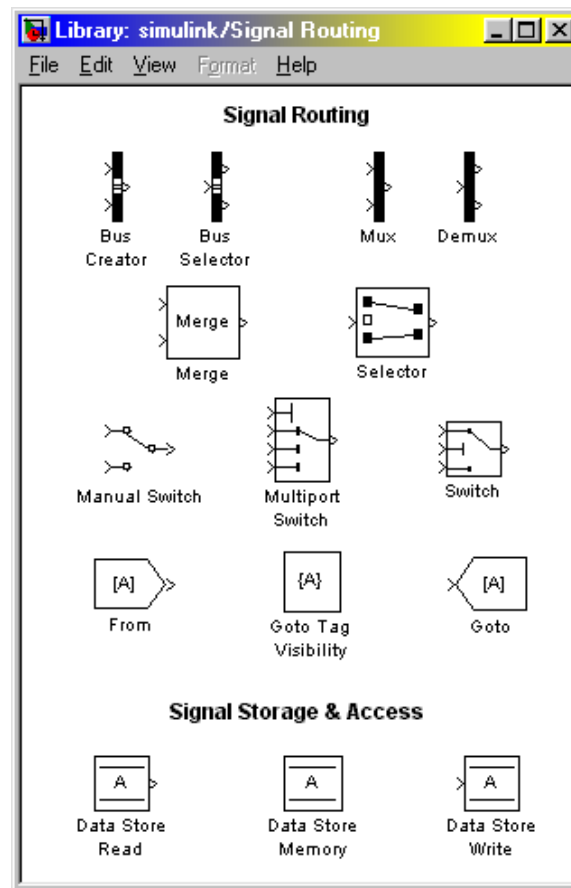


Рис. 14. Вікно розділу Signal Routing бібліотеки SIMULINK

Розділ **Signals Attributes** (рис.15) складається з блоків, що забезпечують або визначення, або змінювання деяких атрибутів сигналу (таких як розмір сигналу (кількість елементів у векторному чи матричному сигналі), тип даних, початкові умови, швидкість передавання даних и т. п.).

Розділ **Discontinuities** (рис. 16) містить 8 елементів, які реалізують різного виду типові кусково-лінійні і розривні залежності виходу від входу.

У розділі **Look Up Tables** (рис. 17) зосереджені блоки, що формують вихідний сигнал за вхідним у відповідності з заданою таблицею значень, здійснюючи лінійну інтерполяцію по цих значеннях.

Розділ **User Defined Functions** (рис. 18) містить блоки, на основі яких користувач може створювати власні S-блоки, які виконуватимуть необхідні йому функції.

У розділі **Model Verification** (рис. 19) містяться блоки, що дозволяють здійснювати перевірку деяких динамічних властивостей S-моделі.

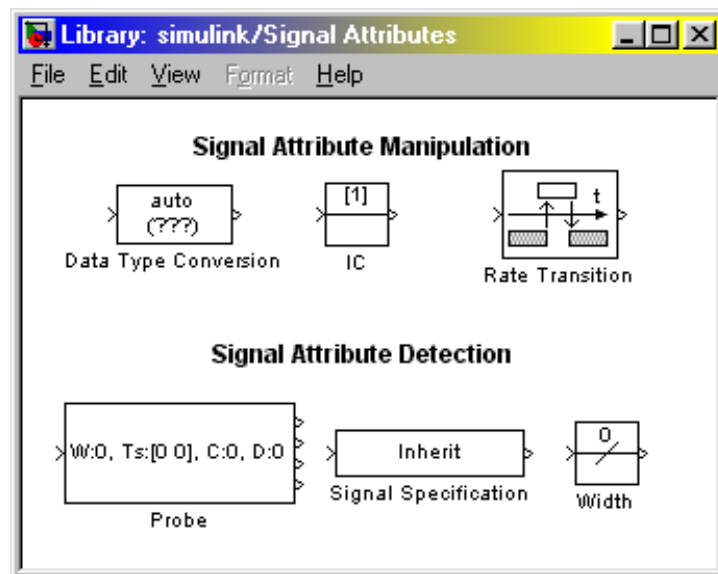


Рис. 15. Вікно розділу Signal Attributes бібліотеки SIMULINK

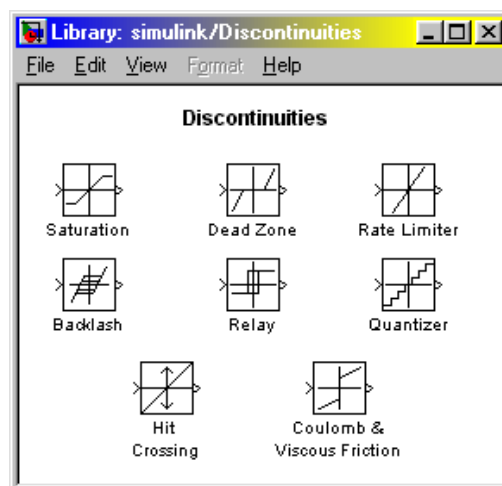


Рис. 16. Вікно розділу Discontinuities бібліотеки SIMULINK

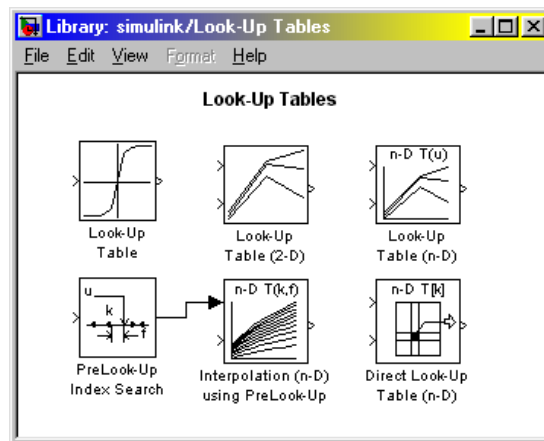


Рис. 17. Вікно розділу Look-Up Tables бібліотеки SIMULINK

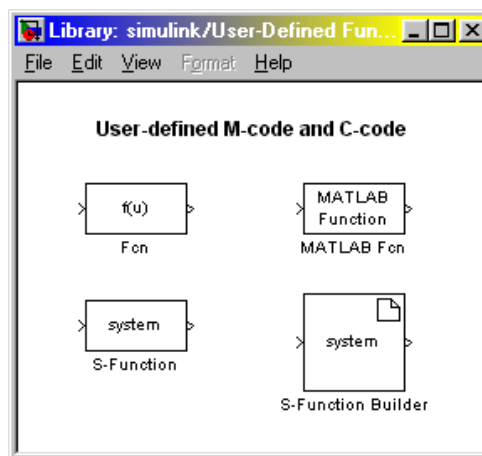


Рис. 18. Вікно розділу User-Defined Functions бібліотеки SIMULINK

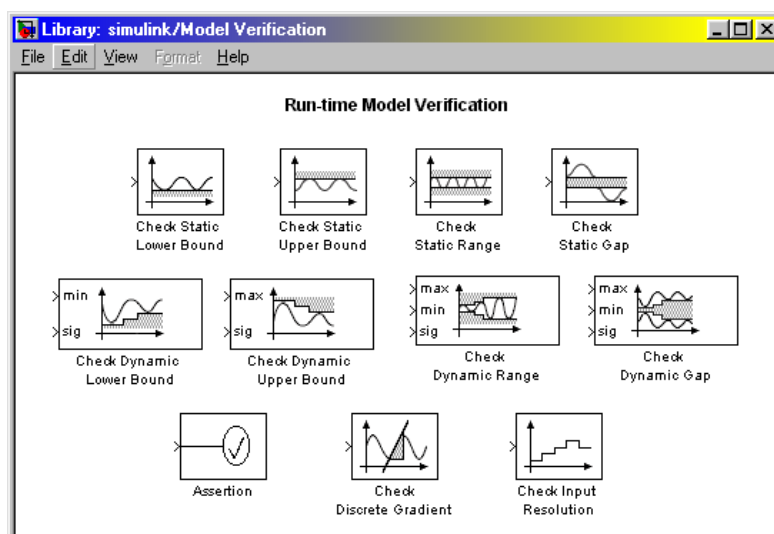


Рис. 19. Вікно розділу Model Verification бібліотеки SIMULINK

Більшість блоків розділу **Ports & Subsystems** (рис. 20) призначено для розробки складених S-моделей, що містять моделі нижчого рівня (підсистеми), і забезпечують встановлення необхідних зв'язків між кількома S-моделями.

ModelWide Utilities (рис. 21) складається з блоків, що дозволяють лінеаризувати нелінійну динамічну модель і оформити документацію до моделі.

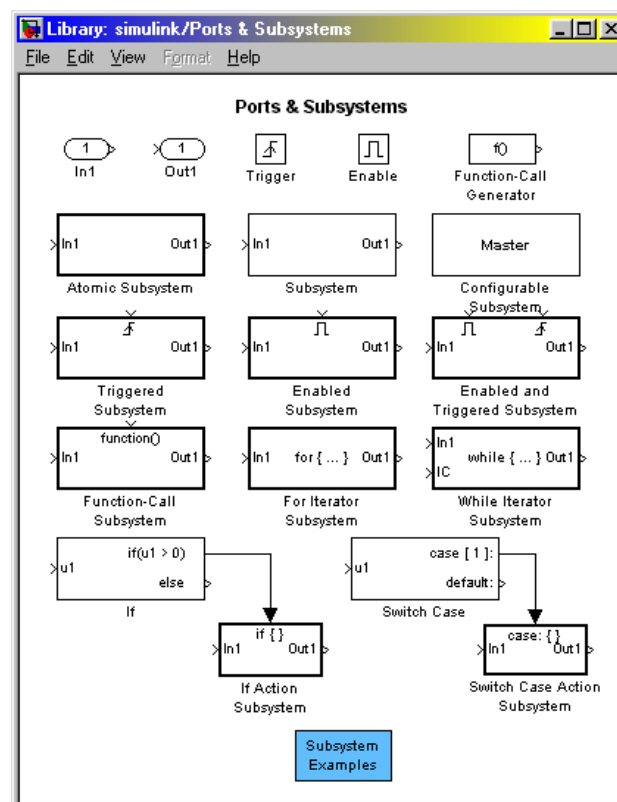


Рис. 20. Вікно розділу Ports & Subsystems бібліотеки SIMULINK

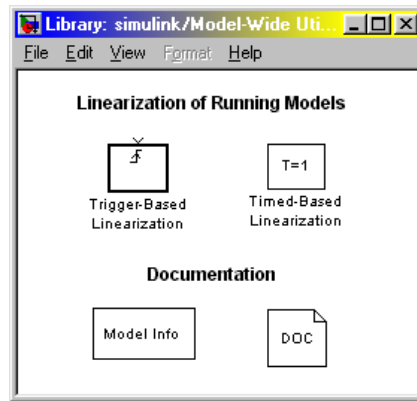


Рис. 21. Вікно розділу Model-Wide Utilities бібліотеки SIMULINK

Чотирнадцятий розділ (рис. 22) - **Blocksets & Toolboxes** (Набори блоків і інструменти) – дає можливість користувачеві використовувати інші додаткові бібліотеки блоків, які увійшли у робочу конфігурацію пакета. П'ятнадцятий розділ **Demos** дозволяє викликати до виконання демонстраційні програми для ілюстрації роботи блоків.

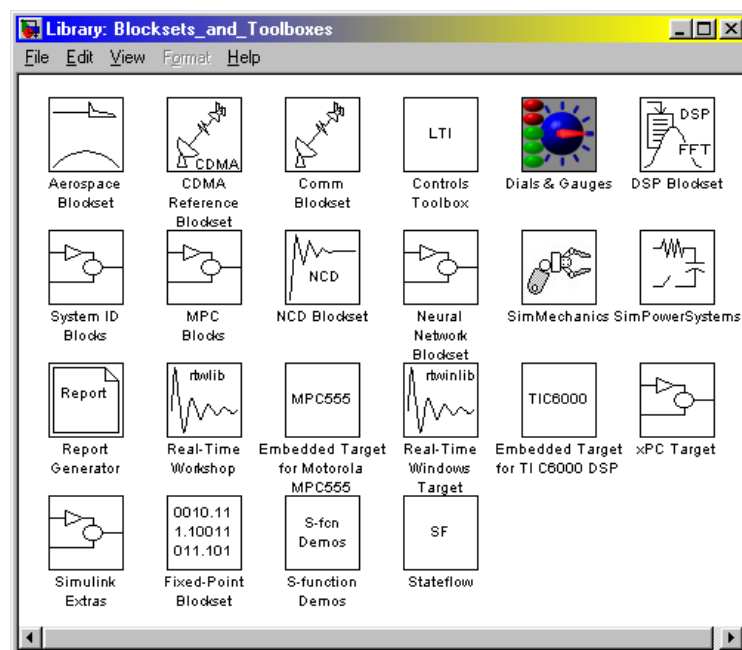


Рис. 22. Вікно розділу Blocksets_and_Toolboxes бібліотеки SIMULINK

1.2.2. СТВОРЕННЯ БЛОК-СХЕМИ S-МОДЕЛІ

Розробка моделей засобами Simulink здійснюється шляхом "складання" блок- схеми моделі на основі блоків бібліотеки SIMULINK у спеціальному вікні блок-схеми (рис. 23), яке викликається через меню File \Rightarrow New \Rightarrow Model у будь-якому з попередніх вікон.

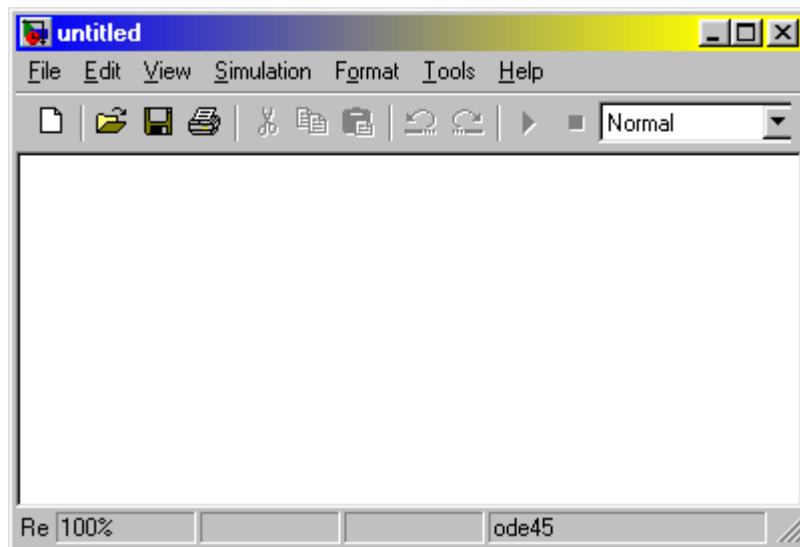


Рис. 23. Порожнє вікно майбутньої блок-схеми

Створення блок-схеми моделі заснована на технології drag-and-drop («перетягни й залиш»). Для цього потрібні блоки з бібліотек Simulink перетягують за допомогою миші у вікно блок-схеми і залишають там. Потім блоки у блок-схемі також за допомогою миші з'єднують один з одним у певній послідовності (вихід попереднього блока з входом наступного блока), утворюючи ланцюг і, у кінцевому підсумку, - схему з'єднання блоків.

Технологія перетягування зображення блока є такою: курсор миші потрібно встановити на зображенні обраного блоку у вікні розділу бібліотеки, потім натиснути ліву клавішу миші і, не відпускаючи її,

передвинути курсор на поле блок-схеми, після чого відпустити клавішу. Аналогічно здійснюється з'єднання у блок-схемі лініями виходів одних блоків зі входами інших блоків: курсор миші підводять до потрібного виходу деякого блоку (при цьому курсор має набути форму хрестика), натискають ліву клавішу і, не відпускаючи її, переміщують курсор до потрібного входу іншого блоку, а потім відпускають клавішу. Якщо з'єднання здійснено вірно, на вході останнього блоку виникне зображення чорної затушованої стрілки.

Фактично блоки бібліотеки являють собою математичні програми перетворення величин, що подаються на вхід блока (вхід блока зображений стрілкою, яка спрямована у бік зображення блока, див. рис. 9), у деякі інші величини, які одержуються на виході блоку (вихід блоку зображений у вигляді стрілки, спрямованої від блока). Процес моделювання полягає у тому, що деякі блоки (джерела) формують величини на їхньому виході як функції аргументу-часу, який змінюється протягом моделювання автоматично, зростаючи з деяким кроком. Ці величини надаються до входів блоків, з'єднаних із джерелом, перетворюються на кожному кроці згідно внутрішньої програми блока у вихідні величини і далі надаються на подальші блоки. У кінцевому підсумку величини, процес змінювання яких з часом є досліджуваним, подаються або до входів оглядових блоків, або до інших блоків-приймачів, які дозволяють спостерігати або зберігати у пам'яті ПК результати моделювання для подальшого їх використання.

Усі величини при такому моделюванні є функціями змінюваного часу. Тому їх прийнято називати сигналами. Отже лінії, що з'єднують блоки, можна розглядати як лінії передавання сигналів, а блоки — як пристрої, що перетворюють вхідний сигнал у вихідний у відповідності з призначенням блока.

Аргумент (значення часу), що використовується при моделюванні, називають модельним часом.

Усі блоки можна поділити на три групи:

- 1) блоки-джерела, які формують (за певним визначеним законом) на виході сигнал у залежності від значення модельного часу;
- 2) блоки-перетворювачі, які перетворюють при кожному значенні модельного часу величину, що подається на вхід блоку, у величину, що одержується на виході блоку;
- 3) блоки-приймачі, які сприймають одержаний на вході сигнал і або виводять його у вигляді графічних залежностей у спеціальних графічних вікнах, або зберігають відповідні масиви чисел у пам'яті.

Будь-яка блок-схема модельованої системи необхідно має включати в себе один або декілька блоків-джерел, що генерують сигнали, які, власне, і спричиняють «рух» модельованої системи, а також один або кілька блоків-приймачів, які дозволяють одержати інформацію про вихідні величини параметрів сигналів цієї системи (побачити результати моделювання).

1.2.3. ВСТАНОВЛЕННЯ ПАРАМЕТРІВ БЛОКІВ

Більшість блоків мають параметри настроювання. Задаючи значення параметра можна конкретизувати функцію, реалізовану даним блоком. Щоб відкрити вікно настроювання параметрів, потрібно перетягнути або скопіювати блок у вікно блок-діаграми й двічі клацнути на ньому. Вікна настроювання параметрів містять коротку характеристику блоку, елементи керування для введення значень параметрів блоку.

Наприклад, вікно настроювання блоку **Sine Wave** має вигляд, наведений на рис. 24. В ньому присутні вікна введення шости параметрів

цього блоку: **Amplitude** (Амплітуда), **Bias** (Стала складова), **Frequency** (Частота), **Phase** (Початкова фаза) і **Sample time** (Дискрет времени).

Значення параметрів настроювання вносяться у вікна введення у таких формах:

- у вигляді чисел;
- у вигляді арифметичних виразів із використанням стандартних вбудованих) функцій MATLAB;
- у формі ідентифікаторів (імен), або алгебричних виразів з використанням ідентифікаторів.

Параметр **Sample time** (Дискрет времени), присутній у вікнах настроювання більшості блоків, встановлюється відмінним від 0 або -1 тільки у випадку, коли на виході блока має бути сформований дискретний за часом сигнал. У цьому випадку величина цього параметра має бути додатною. Вона визначає величину дискрету часу, тобто проміжок модельного часу усередині якого вихідна величина залишається незмінною.

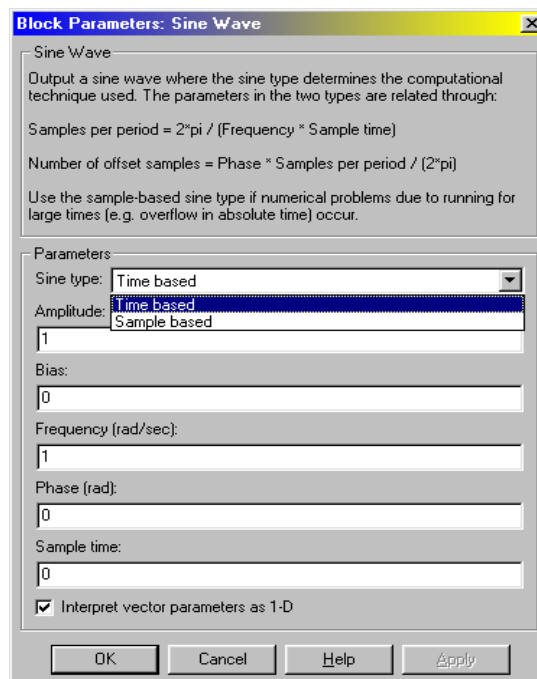


Рис. 24. Вікно настроювання блоку Sine Wave

1.2.4. ВСТАНОВЛЕННЯ ПАРАМЕТРІВ МОДЕЛЮВАННЯ

Після створення блок-схеми моделі можна приступити до моделювання. Але перед тим слід встановити параметри, які визначатимуть сам процес моделювання. Параметри процесу моделювання встановлюються у вікні блок-схеми моделі у спеціальному вікні **Simulation Parameters** (рис. 25), яке виникає на екрані, якщо викликати команду **Simulation Parameters** (Параметри моделювання) з меню **Simulation** (Моделювання).

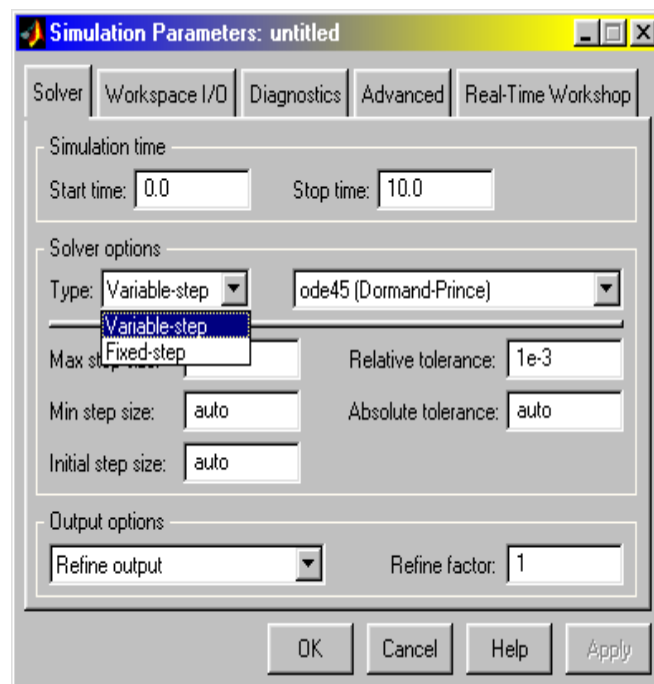


Рис. 25. Вікно Simulations Parameters вікна блок-схеми (Вкладка Solver)

Вікно **Simulation Parameters** має кілька вкладинок. З них найважливішими є вкладинки

- **Solver** (Розв'язувач), показана на рис. 25;
- **Workspace I/O** (Введення-виведення у Робочий простір); вона наведена на рис. 26.

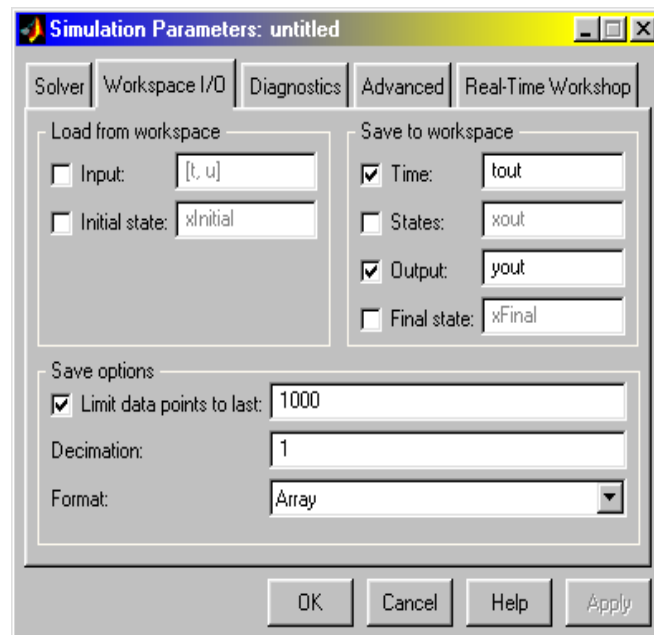


Рис. 26. Вікно Simulations Parameters вікна блок-схеми
(Вкладка Work Space I/O)

Встановлення параметрів моделювання (вкладка Solver)

За допомогою вкладки **Solver** встановлюються параметри, які визначають як будуть змінюватися протягом моделювання значення модельного часу і за яким правилом будуть обчислюватися нові значення змінних стану динамічної системи у чисельній моделі (метод інтегрування диференціальних рівнянь моделі):

- **Start time** (Початковий момент часу) – визначає початкове значення модельного часу у процесі моделювання;
- **Stop time** (Кінцевий момент часу) – визначає кінцеве значення модельного часу;
- **Solver options** (Параметри розв'язувача) – визначає вид розв'язувача (**Variable step** – зі змінюваним кроком чи **Fixed step** – зі сталим кроком – див. рис. 25), параметри кроку змінювання модельного часу і межові значення похибок розв'язувача.

У випадку, коли у списку вікна введення **Type** (рис. 25) обране значення **Fixed step**, вкладинка **Solver** набуває вигляду наведеному на рис. 27.

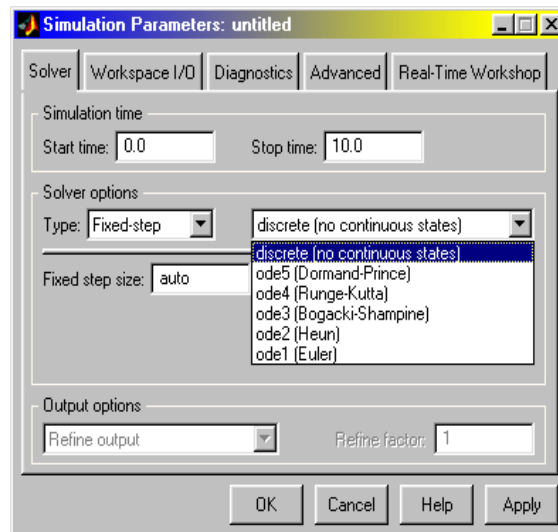


Рис. 27. Вкладка Solver при встановленні Fixed-step

При цьому на вкладинці виникає вікно введення **Fixed step size** (Розмір фіксованого кроку), в яке слід записати бажане значення кроку змінювання модельного часу у процесі моделювання. У списку вікна введення праворуч від вікна введення **Type** (рис. 27) обирається один з пропонованих методів інтегрування з фіксованим кроком.

Якщо модельована система не містить динамічних блоків-ланок (до яких відносяться блоки інтеграторів, диференціюючі блоки і блоки, що задаються у вигляді передатних функцій і рівнянь стану), слід обрати метод **discrete (no continuous states)** (Дискретний (не неперервні стани)).

У випадку, коли встановлено тип розв'язувача **Variable step** зі змінюваним кроком (рис. 25), розмір кроку не встановлюється (він буде визначатися розв'язувачем автоматично, список методів інтегрування пропонується інший, і потрібно додатково встановити межу припустиму відносну похибку моделювання (параметр **Relative tolerance**), або межу припустиму абсолютну похибку (параметр **Absolute tolerance**).

Не важко впевнитися, що якщо обраний розв'язувач з фіксованим кроком, похибка моделювання системою не контролюється (і не гарантується). Єдиною можливістю підвищити точність моделювання у цьому випадку є зменшення значення встановленого фіксованого кроку.

При обранні методу розв'язування зі змінюваним кроком, навпаки, гарантується задана точність моделювання, а кроки модельного часу встановлюються автоматично розв'язувачем на кожному кроці моделювання і тому можуть значно відрізнятися один від одного на протязі інтегрування неконтрольованим чином.

Встановлення параметрів зв'язків з робочим простором системи MATLAB (вкладка The Workspace I/O)

У вкладинці **The Workspace I/O** (див. рис. 26) можна встановити вхідні і вихідні параметри процесу моделювання з використанням робочого простору системи MATLAB. Для цього на вкладинці передбачені три поля

Load from workspace (Завантажити з робочого простору)

Save to workspace (Зберігти у робочому просторі)

Save options (Параметри зберігання)

У полі **Save to workspace** можна встановити ті вихідні величини процесу моделювання, які потрібно записати у робочий простір системи MATLAB. Існують чотири вікна введення з вікнами встановлення прапорців ліворуч від написів:

Time (Час);

States (Стани);

Output (Виход);

Final state (Кінцевий стан).

Якщо прапорець ліворуч від відповідного напису не встановлений, то відповідне вікно введення не є активним (в нього не можна ввести дані).

При встановленні прапорця відповідне вікно введення стає активним. В нього вводяться ймення, під якими зберігатимуться у робочому просторі відповідні дані.

Зазвичай використовуються лише першій і третій вікна введення (див. рис.26, на якому поряд з ними прапорці встановлені). Стан поля **Save to workspace**, наведений на рис. 26 свідчить про те, що усі значення модельного часу при моделюванні будуть записані у вектор **tout**, а значення сигналів, що поступають на блоки **Out** (див. рис. 10) у вікні блок-схеми моделі, зберігатимуться у матриці **yout**. По закінченні процесу моделювання можна використовувати вектор **tout** і матрицю **yout** для огляду одержаних даних і побудови графіків. Звичайно, ймення **tout** і **yout** можна замінити на будь-які інші у вікнах введення поля **Save to workspace**.

Встановлюючи флажок поряд з написом **States** , можна організувати запис поточних значень змінних стану системи у матрицю за йменням **xout** (або іншим, введеним у відповідне вікно праворуч від напису **States**), а, встановивши флажок ліворуч від напису **Final state** – запис кінцевих значень змінних стану у вектор **xFinal**.

Аналогічно у полі **Load from workspace** поряд з написом **Input** можна встановити ймення векторів (за замовчуванням – **t**, **u**) робочого простору MATLAB, з яких будуть зчитуватися дані про значення сигналу (вектор **u**) і відповідні моменти модельного часу (вектор **t**), які при моделюванні будуть використовуватися як заданий сигнал, що спричиняє "рух" моделі при моделюванні. Щоб це було здійснено, у вікні блок-схеми моделі має бути попередньо розміщений блок **In** (див. рис. 9), з виходу якого сигнал має поступати на вхід деякого іншого блоку блок-схеми. Це може стати у нагоді, коли потрібно вивчити вплив на поведінку моделі

деякого сигналу, значення якого одержані експериментально і попередньо записані у файл даних.

У полі **Save options** можна встановити максимальну кількість елементів векторів, що будуть записані у робочий простір (параметр **Limit data points to last** – Гранічна кількість точок у ширину), кількість кроків часу, через яку записуватимуться дані (параметр **Decimation** – Проріджування) і формат запису (параметр **Format**) – **Array** (Масив), **Structure** (Структура) або **Structure with time** (Структура з часом).

1.2.5. ПРОВЕДЕННЯ МОДЕЛЮВАННЯ І ОДЕРЖАННЯ РЕЗУЛЬТАТІВ

Моделювання здійснюється після остаточного складення блок-схеми моделі у вікні цієї блок-схеми. Для цього потрібно викликати команду **Start** з меню **Simulation**. Але попередньо потрібно ввести числові значення усіх ідентифікаторів, що використовувалися при введенні параметрів налаштування блоків, що увійшли у блок-схему, і параметрів моделювання у вікні **Simulation Parameters**.

Наведемо приклад. Нехай створено найпростішу блок-схему (рис.28), що складається з блоку генератора синусоїдального сигналу **Sine Wave** і блока-приймача **Scope**.

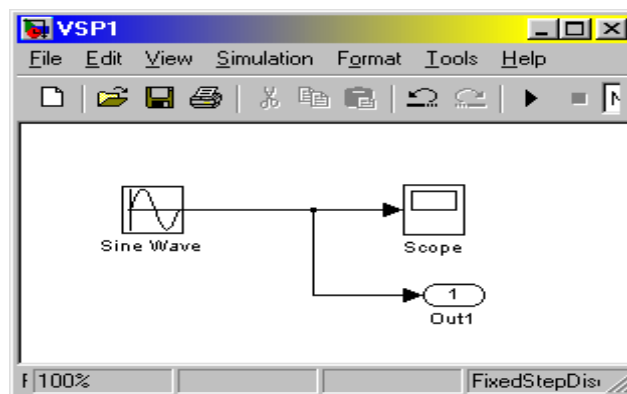


Рис. 28. Блок-схема реєстрації гармонічного сигналу

Крім того, додамо у блок-схему блок **Out1** задля того, щоб одержаний сигнал можна було записати за йменням уout у робочий простір MATLAB.

Параметри моделювання встановимо такими, як подано на рис. 29.

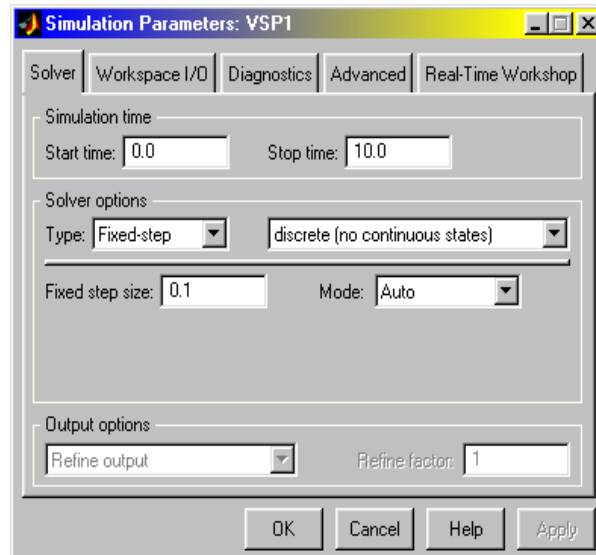


Рис. 29. Параметри моделювання

Запускаючи модель на моделювання, а потім двічі клацаючи на зображенні блоку **Scope**, одержимо графік, наведений на рис. 30.

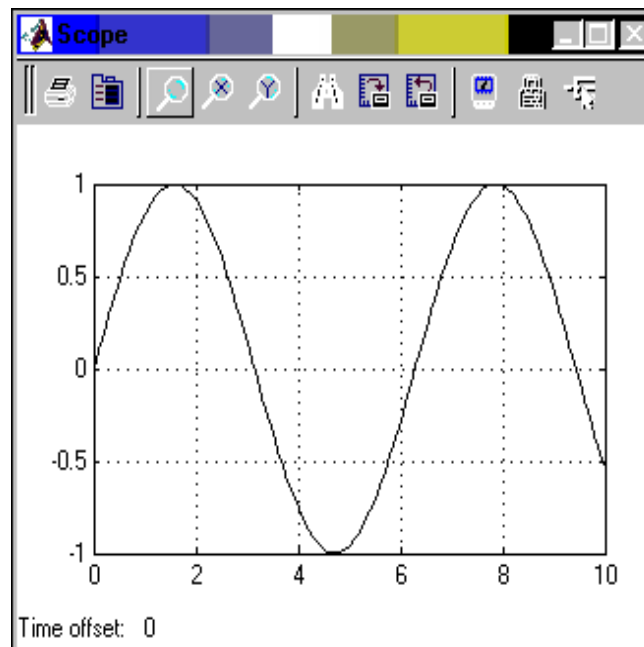


Рис. 30. Графік у вікні блока Scope

Вводячи тепер у командному вікні команди

```
>> plot(tout,yout), grid  
>> title('Графік процесу  $y(t)=\sin(t)$ ')  
>> xlabel('t'), ylabel('y')
```

одержимо у вікні фігури графік тієї самої залежності, наведений на рис. 31.

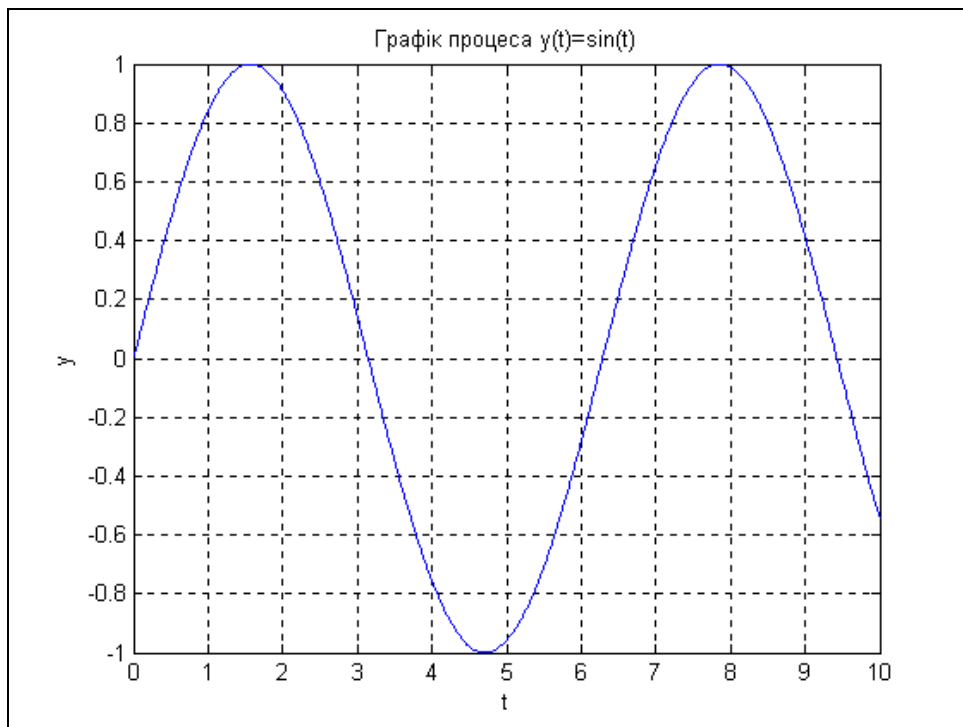


Рис. 31. Графік у вікні Figure

1.3. Контрольні запитання

1. Що розуміють під динамічними системами?
2. У чому полягають основні особливості системи MATLAB, які роблять її найбільш придатною для інженерних досліджень?

3. Що таке "візуальне програмування"? Чому воно називається "візуальним"? Чому воно називається "програмуванням"?
4. Що таке S-модель? У якому вигляді вона зберігається на диску?
5. Що являють собою окремі блоки у пакеті Simulink?
6. З яких етапів складається процес моделювання за допомогою пакету Simulink?
7. Як утворюється блок-схема S-моделі? Які типи блоків мають обов'язково міститися у закінченій блок-схемі?
8. Як дістатися вікна налаштування блоку? У якому виді можна вводити значення параметрів блоку?
9. Присутність яких блоків перетворюють модель у модель динамічної системи?
10. Для чого потрібні блоки-джерела? блоки-приймачі?
11. Що таке "розв'язувачі" ? Як встановлюються розв'язувач та його параметри?
12. Чим вірізняються параметри розв'язувачів з фіксованим кроком від параметрів розв'язувачів зі змінюваним кроком?
13. Як можна впливати на точність моделювання з фіксованим кроком? зі змінюваним кроком?
14. Як здійснити власне моделювання за готовою блок-схемою?
15. Як забезпечити запис результатів моделювання у робочий простір MATLAB?

ЛАБОРАТОРНА РОБОТА №1

ТЕХНОЛОГІЯ МОДЕЛЮВАННЯ У СЕРЕДОВИЩІ SIMULINK

Мета роботи: Навчитись завантажувати MATLAB та SIMULINK, розглянути бібліотеку блоків SIMULINK, засвоїти методику створення моделей засобами SIMULINK, навчитись змінювати параметри моделі та параметрами налаштування окремих блоків.

Завдання для лабораторної роботи

1. Створити блок-модель для:

- виводу в один блок Display розв'язання системи рівнянь, згідно варіанта;
- виводу абсолютних значень коренів рівняння в блок Display;
- виводу найменшого з коренів в блок Scope;
- виводу синуса та косинуса найбільшого з коренів в блок Scope;
- виводу добутку коренів у блок Display.

2. Створити блок-модель для перевірки правильності отриманого рішення.

Всі лінії зв'язку блоків-моделі доповнити назвами, що вказують, які сигнали по них проходять.

Варіанти завдання до лабораторної роботи №1

Варіант 1	Варіант 2	Варіант 3	Варіант 4
$2x-3y+z+14=0$	$3x-4y+2z-2=0$	$2x-4y+2z-26=0$	$2x-3y+z+19=0$
$x+2y+5z+1=0$	$x+6y+2z-4=0$	$9x+2y+3z-47=0$	$x+2y+5z-28=0$
$3x+y-2z+1=0$	$2z+3y-2z-17=0$	$2x+2y-2z+6=0$	$3x+y-2z+33=0$
Варіант 5	Варіант 6	Варіант 7	Варіант 8
$2x-3y+z-2=0$	$3x-7y+5z-77=0$	$4x-5y+3z+8=0$	$5x-7y+3z-84=0$
$x+2y+5z-21=0$	$4x+3y+2z-13=0$	$7x+8y+9z-25=0$	$2x+5y+8z+23=0$
$3x+y-2z+29=0$	$3x+4y-3z+26=0$	$x+2y-3z-19=0$	$3x+6y-2z+49=0$

Варіант 9

Варіант 10

$$8x-7y+2z-38=0$$

$$3x-3y+2z+39=0$$

$$x+4y+5z-37=0$$

$$3x+2y+4z-43=0$$

$$3x+6y-3z+66=0$$

$$4x+10y-8z+15=0$$

Звіт по лабораторній роботі повинен містити блок-модель для рішення системи рівнянь, визначення абсолютних значень, найменшого та найбільшого значень, добутку коренів системи рівнянь, а також блок-модель для перевірки правильності отриманого рішення. Лінії зв'язку блоків-моделей повинні супроводжуватись назвами, що вказують, які сигнали по них проходять, увести в блок-модель коментар із Вашим прізвищем.

Початок роботи в середовищі SIMULINK

Після запуску MATLAB на екрані з'являється заставка, а потім вікно MATLAB див. рис. 1.1. У разі необхідності у спадаючому меню View можна відкрити додаткові вікна, а саме: Command Window (командне вікно), Command History (вікно з командами, що виконувалися раніше), Current Directory (вікно для призначення поточної директорії), Workspace (робоча область). У панелі інструментів є інструмент для запуску SIMULINK.

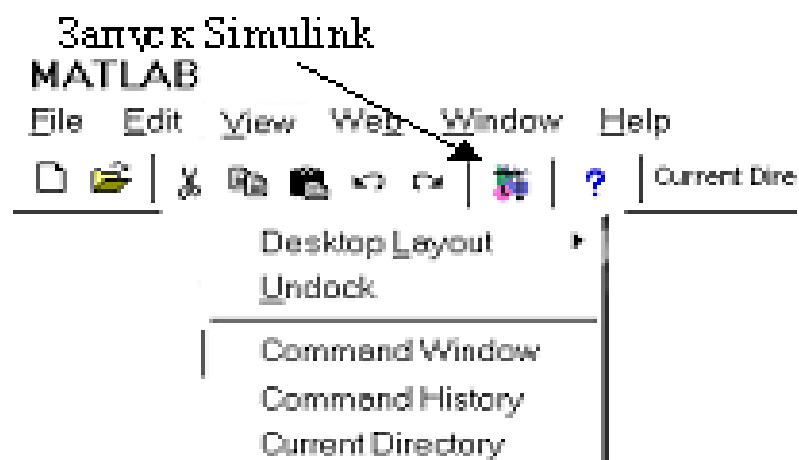


Рис. 1.1. Вікно MATLAB з відкритим спадаючим меню View.

Розробка моделей засобами SIMULINK заснована на технології drag-and-drop («перетягни й залиш»). Для побудови моделі використовуються блоки, які обмінюються між собою даними. Дані можуть бути скалярними величинами, векторами або матрицями довільної розмірності (4—число; [1 4 7]—вектор; $\begin{bmatrix} \sin(1) & 4 & 7 \\ -5 & \pi & 1e-5 \end{bmatrix}$ —матриця).

Робота зі створення блок-моделі полягає в тому, що зображення обраних блоків мишею перетаскуються або копіюються з вікна розділів бібліотеки у вікно зборки моделі, а потім у вікні зборки виходить одних блоків з'єднуються із входами інших блоків.

Для створення нової блок-моделі треба виконати команду New>Model з меню File, після чого відкривається порожнє вікно. Для доступу до блоків, з яких буде створюватися блок-модель, потрібно запустити SIMULINK, клацнувши на відповідній кнопці панелі інструментів вікна MATLAB або ввівши команду `simulink` у рядку командного вікна. При цьому повинно відкритися вікно зі списком розділів бібліотеки SIMULINK.

Блоки SIMULINK для зручності їхнього використання розсортовані в наступні розділи:

Continuous (безперервні системи) — розділ з блоками для моделювання безперервних систем;

Discrete (дискретні) — блоки для моделювання дискретних систем;

Functions & Tables (функції й таблиці);

Math (математичні блоки);

Nonlinear (нелінійні) — блоки для моделювання нелінійних систем;

Signals & Systems (сигнали й системи) — блоки для створення підсистем і керування сигналами;

Sinks (одержувачі) — блоки-одержувачі, використовуються як оглядові вікна для перегляду різних сигналів моделі;

Sources (джерела) — блоки-джерела сигналів (вхідних величин);

Більшість блоків мають параметри настроювання. Задаючи значення параметра можна скорегувати функцію, реалізовану даним блоком. Щоб відкрити вікно настроювання параметрів блоку, потрібно перетягнути або скопіювати блок у вікно блок-діаграми й двічі клацнути на ньому. Вікна настроювання параметрів містять коротку характеристику блоку, елементи керування для введення значень параметрів блоку.

Створіть першу блок-модель для перегляду у вікнах Scope й Display сигналу, сформованого блоком Constant (розділ Sources) у такий спосіб:

1. Виконайте команду New>Model у меню File. Відкриється порожнє безіменне (untitled) вікно для створення нової блок-моделі.
2. Запустіть SIMULINK, клацнувши на відповідній кнопці інструментів або ввівши команду simulink у командному вікні.
3. У розділі Sources знайдіть блок Constant і перетягніть його у вікно untitled.
4. У розділі Sinks (одержувачі) знайдіть блоки Scope (індикатор) і Display, перетягніть їх у вікно untitled.
5. Підведіть покажчик миші до виходу блоку Constant. Покажчик прийме форму хрестика. Утримуючи натиснутою кнопку миші, перемістіть покажчик до входу блоку Scope. Коли хрестик стане подвійним, відпустіть кнопку миші. Між блоками утвориться лінія зв'язку зі стрілкою. Для подачі сигналу на вхід блоку Display підведіть покажчик миші до лінії зв'язку між блоками Constant й Scope, і втримуючи натиснутою клавішу Ctrl, створіть зв'язок аналогічно попередньому.
6. Для запуску процесу моделювання виконайте команду Start (пуск) у меню Simulation або в панелі інструментів натисніть чорний трикутник. Сформований сигнал відобразиться у вікні Display.

7. Для перегляду вмісту вікна Score двічі клацніть на ньому мишкою. Для найкращих умов перегляду сигналу виконаєте автомасштабування (піктограма) або підберіть оптимальний масштаб самі за допомогою контекстного меню, яке викликається натисканням правої кнопки миші.
8. Для створення назви сигналу двічі клацніть на ньому мишкою і у полі, що відкрилося, введіть назву.

Рекомендації з виконання завдання. Для рішення системи алгебраїчних рівнянь використовується блок алгебраїчний корінь Algebraic Constraint (розділ Math). Пошук коренів виконується методом послідовних наближень, тобто носить ітераційний характер. Початкове значення ітераційного процесу задається як значення параметра Initial guess (початкове наближення) у вікні налаштування блоку. На рис. 1.3 наведена блок-модель рішення системи рівнянь:

$$3z_1 - z_2 - 17 = 0$$

$$-5z_1 - z_2 + 7 = 0$$

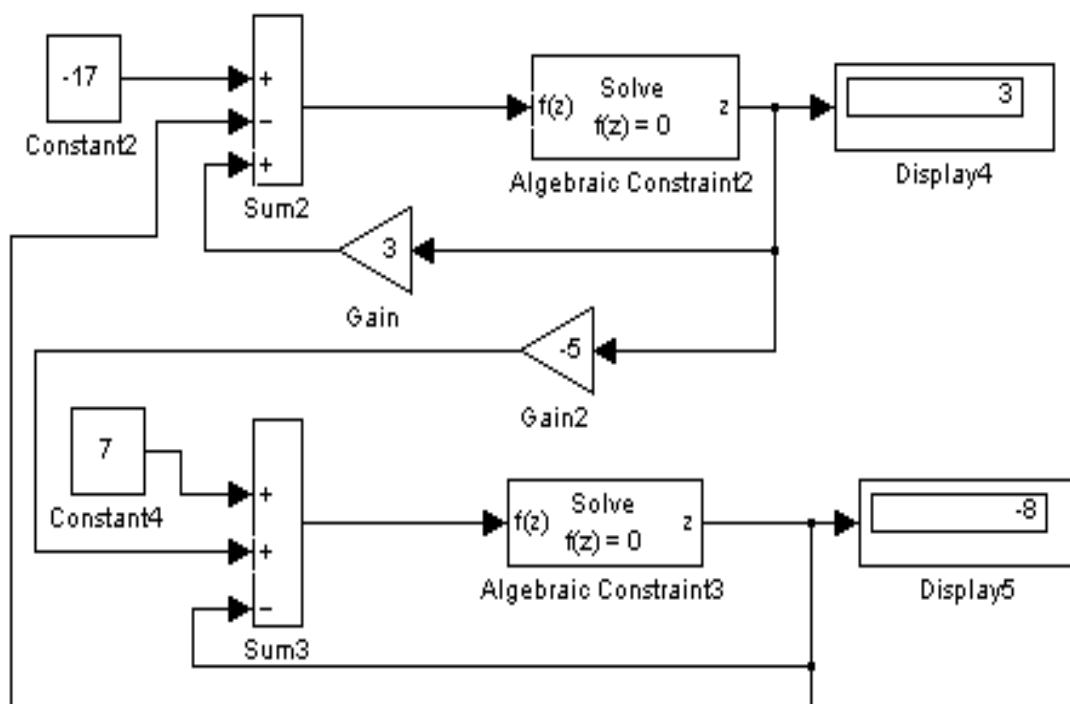


Рис. 1.2. Рішення системи алгебраїчних рівнянь

Контрольні запитання

1. Як створити нову блок-модель?
2. Що таке параметри налаштування блоків, для чого вони призначені?
3. Поясніть призначення блоків Constant, Scope, Display, Algebraic Constraint які параметри налаштування цих блоків?
4. Які блоки використовуються для обчислення абсолютного значення, найменшого, найбільшого, добутку з кількох чисел, синуса та косинуса.

ЛАБОРАТОРНА РОБОТА № 2

ФОРМУВАННЯ ТА ДОСЛІДЖЕННЯ СТАНДАРТНИХ ВХІДНИХ ІНФОРМАЦІЙНИХ СИГНАЛІВ, ЩО ВИКОРИСТОВУЮТЬСЯ ПРИ ВИЗНАЧЕННІ ДИНАМІЧНИХ ХАРАКТЕРИСТИК ПРИЛАДІВ

Мета роботи: Ознайомитись з призначенням та параметрами налаштування блоків генераторів (Sources) та приймачів (Sinks) сигналів бібліотеки SIMULINK. Засвоїти об'єднання типових, найбільш уживаних в інженерній практиці, сигналів у єдиний векторний сигнал, методи формування сигналів у вигляді випадкових процесів з заданими характеристиками і з заданими спектральними характеристиками

Завдання до лабораторної роботи.

1. Створити блоків-моделі для виводу вихідних сигналів різних генераторів у вікно Scope (розділ Sinks). Провести моделювання для значень параметрів налаштування блоків заданих за замовчуванням, а також заданих згідно варіанта. Зробити висновки про вплив параметрів налаштування блоків на вид вихідного сигналу. Розглянути наступні блоки генератори сигналів:

1.1. Генератор гармонійних коливань за допомогою блоку SineWave (розділ Sources).

№ варіанта	1	2	3	4	5	6	7	8	9	10
Amplitude (амплітуда)	1	2	3	4	1	2	3	4	1	2
Bias (переміщення синусоїди по осі ординат)	1	3	2	1	50	5	8	16	20	30
Frequency (частота), радіан/с	4	3	2	1	1	2	3	4	3	4
Phase (початкова фаза), радіан	1	5	1	1	20	2	7	12	25	30
Sample time (крок генерування сигналу даним блоком, при значенні (-1) буде рівний кроку зміни модельного часу, заданому в параметрах Simulation parameters)	Виконати моделювання для значень (-1); 0,1; 0,2; 1; 2 і зробити висновки про вплив параметра на сигнал, що генерується									

1.2. Генератор нормально розподілених випадкових чисел за допомогою блоку Random Number (розділ Sources).

№ варіанта	1	2	3	4	5	6	7	8	9	10
Mean(середнє значення)	1	2	3	4	1	2	3	4	1	2
Variance(дисперсія-середнє квадратичне відхилення від середнього)	1	3	2	10	50	5	8	16	20	30
Initial seed(початковий параметр генерації, який необхідно змінювати для одержання нової послідовності випадкових чисел)	4	3	2	1	1	2	3	4	3	4

Sample time (крок генерування сигналу даним блоком, при значенні (-1) буде рівний кроку зміни модельного часу, заданому в параметрах Simulation parameters)	Виконати моделювання для значень (-1); 0,1; 0,2; 1; 2 і зробити висновки про вплив параметра на сигнал, що генерується
---	--

1.3. Генератор типового впливу у вигляді одиничної сходишки за допомогою блоку Step (розділ Sources).

№ варіанта	1	2	3	4	5	6	7	8	9	10
Step time (час генерування початкового значення сигналу)	1	2	3	4	5	1	2	3	4	5
Initial value(початкове значення сигналу)	5	4	3	2	1	1	2	3	4	5
Final value(кінцеве значення сигналу)	6	5	4	3	2	2	3	4	5	6
Sample time (крок генерування сигналу даним блоком, при значенні (-1) буде рівний кроку зміни модельного часу, заданому в параметрах Simulation parameters)										

1.4. Генератор постійно наростаючого сигналу за допомогою блоку Ramp.

№ варіанта	1	2	3	4	5	6	7	8	9	10
Stope (крутість нахилу прямої до осі часу)	9	8	9	4	5	9	8	9	4	50
Start time (час до початку наростання сигналу)	5	4	3	2	1	1	2	3	4	5
Initial output (початкове значення сигналу)	6	5	4	3	2	2	3	4	5	6

1.5. Генератор безперервного сигналу часу за допомогою блоку Clock (розділ Sources) .

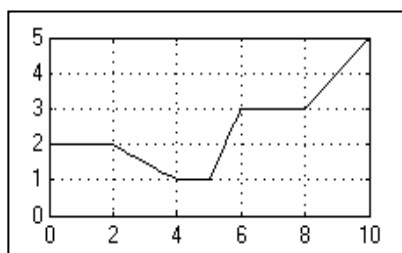
№ варіанта	1	2	3	4	5	6	7	8	9	10
Display time (виводити/не виводити час у піктограму блоку)	9	8	9	4	5	9	8	9	4	50
Decimation (крок генерування вихідного сигналу дорівнює добутку цього параметра на крок зміни модельного часу, заданому в параметрах Simulation parameters)										

1.6. Генератор сигналів довільної форми за допомогою блоку Signal Generator (розділ Sources).

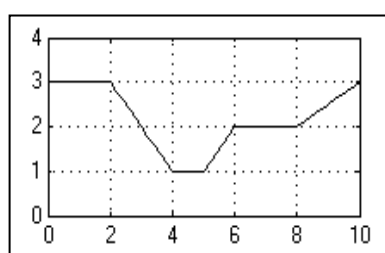
№ варіанта	1	2	3	4	5	6	7	8	9	10
Amplitude	1	2	3	4	5	4	3	2	4	3
Frequency (частота коливань)	1	0,5	0,5	1	1	1	1	1	1	1
Тип періодичного процесу	Сину-соїда		Прямо-кутник		Трик-утник		Випадкові коливання			
Units (одиниця виміру частоти)	Hertz				Rad/Sec					

1.7. Генератор повторюваних сигналів заданої в параметрах блоку форми за допомогою блоку Repeating Sequence (розділ Sources). Форму сигналу, що повторюється забезпечити згідно варіанту.

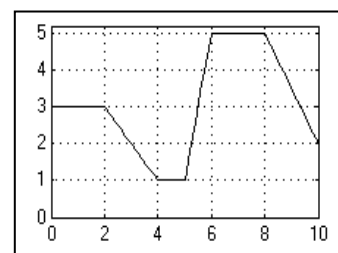
Варіант 1.



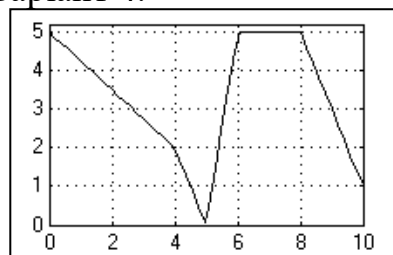
Варіант 2.



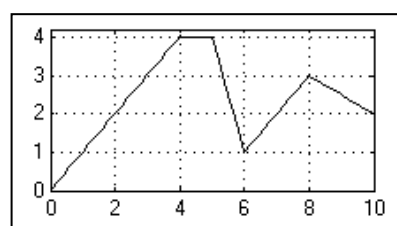
Варіант 3.



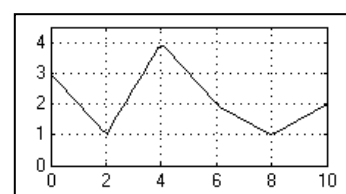
Варіант 4.



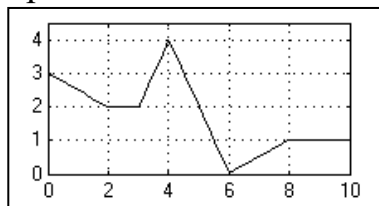
Варіант 5.



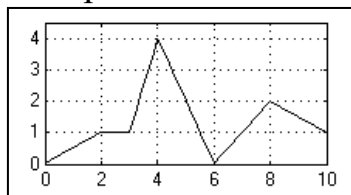
Варіант 6.



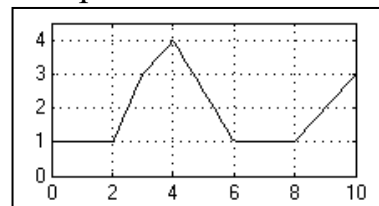
Варіант 7.



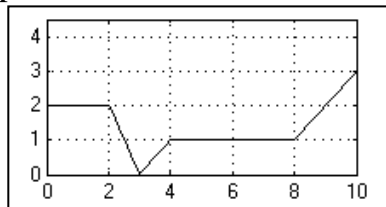
Варіант 8.



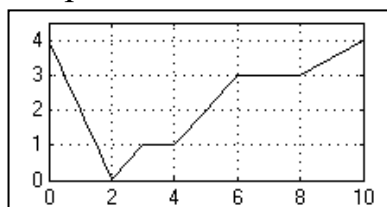
Варіант 9.



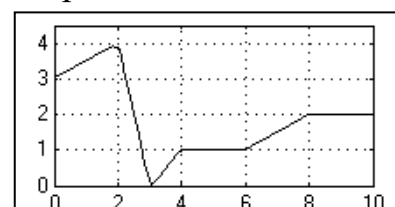
Варіант 10.



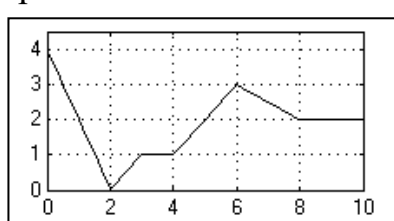
Варіант 11.



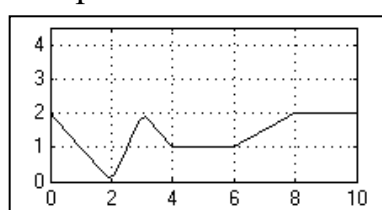
Варіант 12.



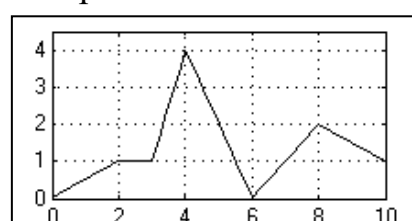
Варіант 13.



Варіант 14.



Варіант 15.



1.8. Генератор періодичного сигналу прямокутної форми за допомогою блоку Pulse Generator (розділ Sources).

№ варіанта	1	2	3	4	5	6	7	8	9	10
Amplitude(амплітуда)	9	8	9	4	5	9	8	9	4	9
Period(період), с	5	4	3	2	1	1	2	3	4	5
Pulse Width(ширина імпульсу в % від періоду)	30%			25%			75%			
Phase delay(час початку генерування імпульсів), с	5	4	3	2	1	1	2	3	4	5

1.9. Генератора синусоїдальних коливань одиничної амплітуди й змінної частоти, що лінійно змінюється в часі за допомогою блоку ChirpSignal.

№ варіанта	1	2	3	4	5	6	7	8	9	10
Initial Frequency(початкова частота), Гц	1	5	1	6	1	5	1	1	1	1

Targed time (наступний момент часу),с	50	50	70	70	50	50	80	80	90	90
Frequency at target time(частота в наступний момент часу), Гц	5	1	6	1	5	1	5	6	5	6

2. Робота з робочою областю MATLAB Workspace.

2.1. Побудуйте графік функції, аргумент і значення функції, якої задані у текстовому файлі MS-DOS. Варіанти завдань візьміть із пункту 1.7.

2.2. Створити блок-модель виводу графіка функції $f(x)$ у вікно Scope з робочої області MATLAB за допомогою блоку From Workspace (розділ Sources). Значення часу t і вихідного сигналу $f(x)$ увести в Workspace з командного рядка MATLAB у вигляді значень двох змінних. У звіті по лабораторній роботі приведіть вид цих змінних.

№ варіанта	1	2	3	4	5	6	7	8	9	10
Функція $f(x)$	$\sin(u)$	$\sin(5u)$	$\cos(u)$	$3\cos(u)$	$\tan(5u)$	$2\sin(3u)$	$2\tan(6u)$	$2+3\sin(6u)$	$2\tan(u)+2\sin(3u)$	$5\sin(10u)$
Інтервал зміни часу t	0–1	0–2	1–5	2–7	1–5	1–3	1–3	2–5	5–6	5–9
Крок зміни часу	0.2	0.1	0.3	0.1	0.2	0.5	0.2	0.2	0.5	0.5
Інтервал зміни функції $f(x)$	0–1	0–2	1–5	2–7	1–5	1–3	1–3	2–5	5–6	5–9
Крок зміни функції	0.2	0.1	0.3	0.1	0.2	0.5	0.2	0.2	0.5	0.5

2.3. Створити блок-модель виводу графіків функцій у вікно Scope з робочої області MATLAB за допомогою блоку From Workspace (розділ Sources). Значення часу й вихідних сигналів увести в Workspace з командного рядка MATLAB у вигляді значення однієї змінної. У звіті по лабораторній роботі приведіть вид цієї змінної.

№ варіанта	1	2	3	4	5	6	7	8	9	10
Функції $f(t)$ і $f_1(t)$	$\sin(t)$ і $\sin(5t)$	$\sin(5t)$ і $\cos(t)$	$\cos(t)$ і $3\cos(t)$	$3\cos(t)$ і $\tan(5t)$	$\tan(5t)$ і $2\sin(3t)$	$2\sin(3t)$ і $2\tan(6t)$	$2\tan(6t)$ і $3\sin(6t)$	$\sin(6t)$ і $2\tan(t)$	$2\tan(t)$ і $2\sin(3t)$	$5\sin(10)$ і $2\tan(t)$
Інтервал по t	0–1	0–2	1–5	2–7	1–5	1–3	1–3	2–5	5–6	5–9
Крок зміни t	0.2	0.1	0.3	0.1	0.2	0.5	0.2	0.2	0.5	0.5

2.4. Створити блок-модель виводу результатів моделювання завдання 2.1. в змінну робочої області з Вашим ім'ям за допомогою блоку To Workspace (розділ Sinks) і у файл із Вашим ім'ям за допомогою блоку To File.

2.5. У командному вікні MATLAB переглянете й запишіть значення змінної робочої області для різних форматів збереження даних Save format, які можна міняти в параметрах блоку To Workspace.

3. Об'єднання, підсумовування і множення сигналів. Створити блок-модель для виведення наступних вихідних сигналів у вікно Scope:

- двох генераторів ступінчатого сигналу Step та генератора гармонічних коливань Sine Wave у вигляді єдиного векторного сигналу;
- суми вихідних сигналів двох генераторів ступінчатого сигналу Step;
- суми вихідних сигналів двох генераторів ступінчатого сигналу Step та генератора гармонічних коливань Sine Wave;
- добутку суми двох генераторів ступінчатого сигналу Step та генератора гармонічних коливань Sine Wave.

Рекомендації до виконання пункту 3 завдання дивись далі.

Варіанти завдань до пункту 3.

Параметри блоку генератора гармонічних коливань Sine Wave.

№ варіанта	1	2	3	4	5	6	7	8	9	10
Amplitude (амплітуда)	0.5	1	2.5	3	3.5	3	2.5	2	1.5	0.8
Frequency(частота), радіан/с	4*pi									

Параметри першого блоку генератора ступінчатого сигналу Step

№ варіанта	1	2	3	4	5	6	7	8	9	10
Step time (час генерування початкового значення сигналу)	1	2	3	4	5	1	2	3	4	5
Final value(кінцеве значення сигналу)	1	1.2	2.8	3.5	4	3.6	2.9	2.4	1.7	1.3

Параметри другого блоку генератора ступінчатого сигналу Step

№ варіанта	1	2	3	4	5	6	7	8	9	10
Step time (час генерування початкового значення сигналу)	6	7	8	9	10	6	7	8	9	10
Final value(кінцеве значення сигналу)	-1	-1.2	-2.8	-3.5	-4	-3.6	-2.9	-2.4	-1.7	-1.3

4. **Генерування випадкових процесів.** Створити блок-схему формування випадкових сигналів яка складається з двох стандартних блоків **Random Number** і **Band-Limited White Noise**, двох однакових формуючих фільтрів на ґрунті блоків **Transfer Fcn**, двох оглядових блоків **Scope** та чотирьох вихідних портів **Out**. В блоці **Transfer Fcn** з розділу **Continuous** бібліотеки **Simulink** задати передатну функцію коливальної ланки такого вигляду:

$$W(s) = \frac{1}{\frac{1}{lam^2} s^2 + \frac{2 * mu}{lam} s + 1}$$

Рекомендації до виконання пункту 4 завдання дивись далі.

Варіанти завдань до пункту 4.

Параметри блоку Random Number

№ варіанта	від 1 до 10
Mean (середнє значення сигналу)	0
Variance (середнє квадратичне відхилення сигналу від його середнього значення)	1
Initial seed (- код бази генерації, який визначає псевдовипадкову послідовність чисел)	0
Sample time (крок генерування сигналу даним блоком)	-1

Параметри блоку Band-Limited White Noise

№ варіанта	1	2	3	4	5	6	7	8	9	10
Noise power (потужність білого шуму)	[0.1]									
Sample time (дискрет часу (визначає верхнє значення частоти процесу))	0.1	0.2	0.3	0.4	0.5	0.4	0.2	0.3	0.1	0.5
Seed (початкове значення бази генератора випадкових чисел)	23341									

5. **Робота з блоком XYGraph.** Створити блок-модель виводу у вікно XYGraph графіка функції $f(u)$, а у блок Display – числового значення функції при $u=a$. У те ж вікно XYGraph вивести горизонтальну й вертикальну лінії на перетинанні, яких повинна перебувати точка $f(a)$ див. рис. 2.1. Рекомендації до виконання пункту 5 завдання дивись далі.

№ варіанта	1	2	3	4	5	6	7	8	9	10
Функція $f(u)$	$\sin(u)$	$\cos(u)$	$\sin(u)$	$\cos(u)$	$\sin(u)$	$\cos(u)$	$\sin(u)$	$\cos(u)$	$\sin(u)$	$\cos(u)$
Час моделювання	30	30	30	30	30	30	30	30	30	30
Значення a	1	1	3	2	4	4	6	5	7	7

Звіт по виконанню лабораторної роботи повинні містити:

1. Графіки вихідних сигналів генераторів.
2. Зміст текстового файлу та змінної робочої області MATLAB по даним, з якого побудовано графік функції, аналогічний графіку завданню із завдання 1.7 (завдання 2.1)
3. Командні рядки MATLAB для введення в Workspace значень часу t і вихідного сигналу $f(x)$ у вигляді однієї та двох змінних. Вигляд цих змінних. Варіанти форматів збереження даних у робочу область MATLAB Workspace.
4. Блок-діаграму для об'єднання, підсумовування і множення сигналів та графіки отримані при моделюванні.
5. Блок-діаграму для генерування випадкових процесів сигналів та графіки отримані при моделюванні.
6. Блок-діаграму для виводу $f(a)$ на Display та у вікні XYGraph графіка $f(u)$ з горизонтальною, вертикальною лініями, що перетинаються в точці $f(a)$.

Рекомендації до виконання пункту 2. Робота з робочою областю MATLAB Workspace.

Для побудови графіка функції, аргумент і значення функції, якої задані у текстовому файлі засобами, наприклад, текстового редактору Блокнот, створіть текстовий файл MS-DOS, у якому запишіть два рядки цифр, що розділені пробілом. У першому рядку запишіть значення аргументу, а у другому рядку – відповідні значення функції.

Для зчитування даних записаних у файлі в змінну робочої області MATLAB Workspace виконайте послідовність команд Файл > Імпорт даних... з меню MATLAB та відкрийте створений Вами раніше файл. При цьому з'явиться вікно, як на рис.2.1.

What column separator does your data use?

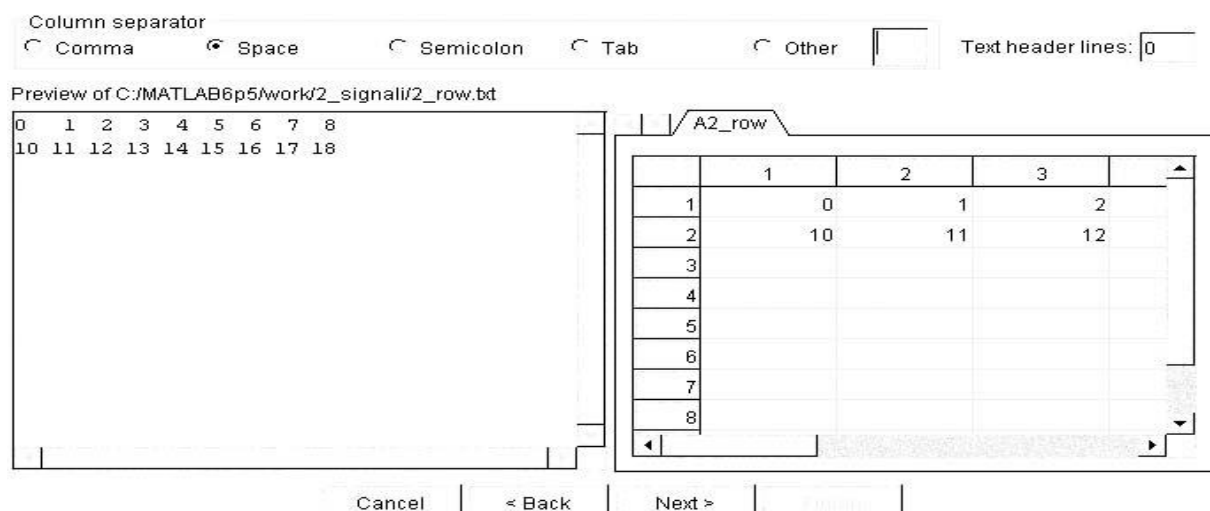


Рис. 2.1. Вікно Import Wizard для імпорту значень аргументу та функції з текстового файлу у змінну робочої області MATLAB Workspace.

Проконтролюйте чи вірно відображені числові значення аргументу та функції, зверніть увагу на ім'я змінної, що буде створена у Workspace (для рис.2.1. це змінна A2_row), натисніть Next і Finish. Для перегляду змісту робочої області виконайте послідовність команд Вид > Workspace з меню MATLAB, завдяки чому відкриється вікно, у якому перераховані всі змінні робочої області, див рис. 2.2.

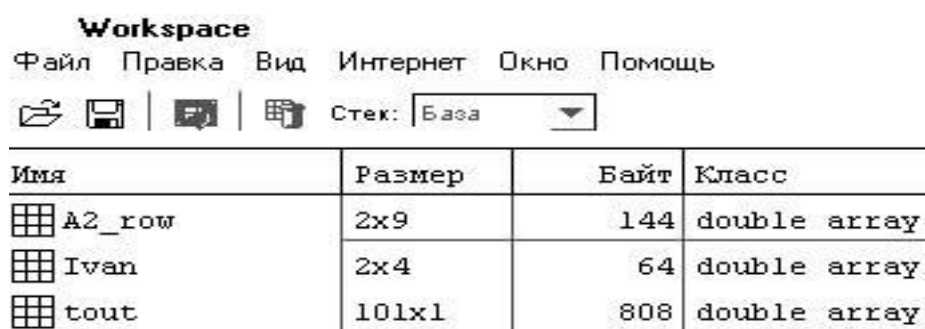


Рис. 2.2. Вікно робочої області MATLAB Workspace.

Клацнувши мишкою на змінній імені змінної A2_row, відкриємо вікно у якому можна редагувати значення змінної, див рис. 2.3.

Array Editor: A2_row

Файл Правка Вид Інтернет Окно Помощь

Числовой формат: shortG Размер: 2 по 9

	1	2	3	4	5	6	7	8
1	0	1	2	3	4	5	6	7
2	10	11	12	13	14	15	16	17

Рис. 2.3. Редагування значень змінної A2_row з робочої області MATLAB Workspace.

Для збереження значень змінної у файлі відкрийте контекстне меню цієї змінної (клацнути правою кнопкою миші на A2_row) і виберіть Сохранить выбранное как..., введіть ім'я файлу і збережіть файл у підкаталозі Work каталогу MATLAB. Для побудови графіка функції створіть блок-модель аналогічну показаній на рис. 2.4.



Рис. 2.4. Побудови графіка функції по даним змінної A2_row з робочої області MATLAB Workspace та даним із файлу A2_row.mat.

Перегляньте отримані графіки і зверніть увагу, що по даним із змінної робочої області A2_row побудовано декілька графіків замість одного. Це пояснюється тим, що у змінної робочої області повинно бути два стовпчики, а саме: стовпчик із значення аргументу і стовпчик із відповідними значеннями функції. У файлі данні відповідно розташовані у рядках. Для редагування даних записаних у файлі прочитайте їх у змінну

робочої області MATLAB Workspace (Файл > Импорт данных...) і відкрийте данні змінної.

Зверніть увагу, що блоки From Workspace і To Workspace не можуть працювати в одній моделі, тому що необхідно розділити їхню роботу в часі (аналогічно From File й To File).

Для графіка функції $f(x)$ у вікно Scope з робочої області MATLAB використайте блок From Workspace. Якщо час і значення функції задавати у вигляді двох матриць, то в командному рядку MATLAB треба виконати, наприклад, команди:

`>>T=[0:0.1:10]'` – вектор значень моментів часу, тобто час міняється від 0 до 10 із кроком 0.1; символ `'` – вказує завдання вектора-стовпця

`>>U=[sin(T) cos(T)]` – матриця значень вихідних сигналів у відповідні моменти часу, а параметр блоку **Data** буде – **[T,U]**.

Якщо час і значення функції задавати у вигляді однієї матриць, то в командному рядку MATLAB треба виконати, наприклад, команду:

`>>U=[0:0.1:10; sin(0:0.1:10)]'` – одна матриця, 1-ий стовпчик якої це значення часу, а інші - значення сигналів, тоді значення **Data** буде – **[U]**.

Рекомендації до виконання пункту 5. Об'єднання, підсумовування і множення сигналів

Для об'єднання кількох сигналів у єдиний векторний сигнал використовується блок **Mux**, який утворює з кількох сигналів, що подаються на входні порти блоку, єдиний векторний сигнал, утворюючи при цьому ніби "шину", всередині якої кожний сигнал передається, відокремленим від інших сигналів, не змішуючись з ними, але у складі з ними як єдине ціле. Таке об'єднання сигналів корисне, для подальшого застосування до кожного з сигналів шини однакових операцій їхнього перетворення. Наприклад, подаючи векторний сигнал на вхід блоку Scope,

можна одержати у його графічному вікні зображення одночасно графіків усіх сигналів, що складають цей векторний сигнал.

Для ілюстрації розглянемо блок-схему, що наведена на рис. 2.5. Вона складається з трьох блоків генерування сигналів (два генерують ступінчасті сигнали, а третій – сигнал синусоїдальної форми) одного оглядового блоку типа **Scope** і блоку **Mux**. Для забезпечення виведення результатів моделювання у робочий простір Matlab додано вихідний порт **Out**. Значення параметрів налаштування блоків, що генерують сигнали, подані на рис. 2.6. і 2.7.

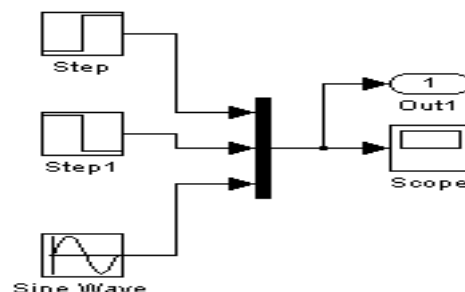


Рис. 2.5. Блок-схема об'єднання сигналів у єдиний векторний сигнал

Block Parameters: Step

Step
Output a step.
Parameters
Step time:
5
Initial value:
0
Final value:
1.2
Sample time:
0
<input checked="" type="checkbox"/> Interpret vector parameters as 1-D
<input checked="" type="checkbox"/> Enable zero crossing detection

Block Parameters: Step1

Step
Output a step.
Parameters
Step time:
10
Initial value:
0
Final value:
-1.2
Sample time:
0
<input checked="" type="checkbox"/> Interpret vector parameters as 1-D
<input checked="" type="checkbox"/> Enable zero crossing detection

Рис. 2.6. Вікна налаштування блоків генерування ступінчастих сигналів

Block Parameters: Sine Wave

- Sine Wave

Output a sine wave where the sine type determines the computational technique used. The parameters in the two types are related through:

$$\text{Samples per period} = 2\pi / (\text{Frequency} * \text{Sample time})$$

$$\text{Number of offset samples} = \text{Phase} * \text{Samples per period} / (2\pi)$$

Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.

- Parameters

Sine type: Time based

Amplitude:

0.5

Bias:

0

Frequency (rad/sec):

4*pi

Phase (rad):

0

Sample time:

0

☒ Interpret vector parameters as 1-D

Block Parameters: Mux

- Mux

Multiplex scalar, vector, or matrix signals into a bus.

- Parameters

Number of inputs:

3

Display option:

bar

Рис. 2.7. Вікна налаштування блока Sine Wave і блока Mux

Здійснивши запуск цієї моделі, можна побачити графіки трьох процесів в оглядовому вікні блока **Scope**. Щоб отримати графіки сигналів, у командному вікні Matlab введемо послідовність операторів:

```
plot(tout,yout(:,1),tout,yout(:,1),'.',tout,yout(:,2),tout,yout(:,2),'*',tout,yout(:,3)),  
grid  
title('Об'єднання трьох сигналів у єдиний векторний сигнал')  
xlabel('t')  
legend('приступок 1','приступок 2','синусоїда',0)
```

В результаті у графічному вікні одержимо графік, поданий на рис. 2.8.

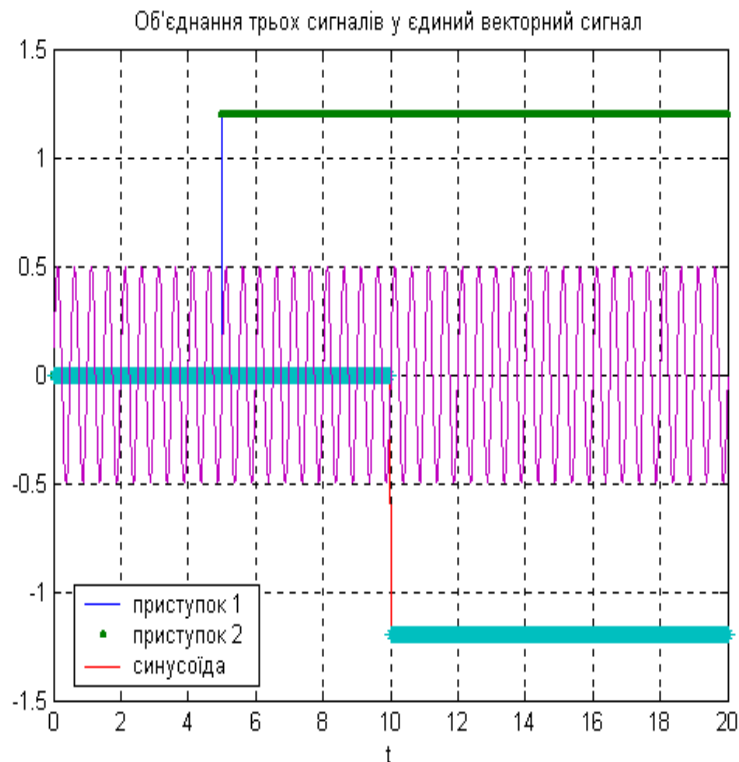


Рис. 2.8. Результат об'єднання трьох сигналів за допомогою блока Mux

Часто виникає потреба утворити сигнали більш складного вигляду, ніж утворювані стандартними блоками бібліотеки Simulink. Такі сигнали іноді можна подати як суму або добуток кількох сигналів типових форм за допомогою блоків **Sum** і **Product** з розділу **Math operations**.

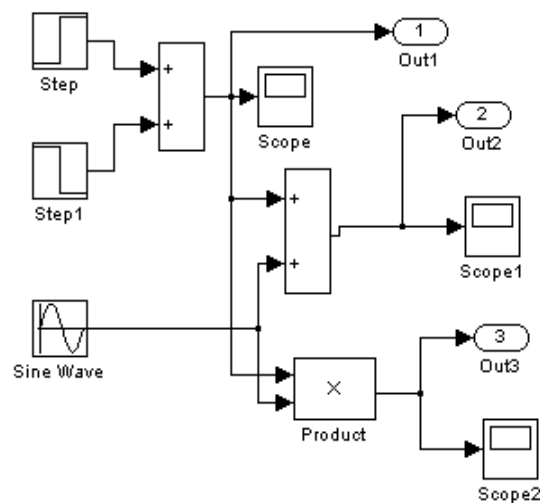


Рис. 2.9. Блок-схема ілюстрування складання і множення сигналів

Блок-схема (рис. 2.9) складається з трьох блоків генерування сигналів та оглядових блоків **Scope**, двох блоків **Sum** підсумовування сигналів і блока множення **Product**. Для забезпечення виведення результатів моделювання у робочий простір Matlab використано три вихідних порти **Out**. Після здійснення моделювання одержимо в оглядових вікнах графіки сигналів.

Щоб отримати графік сигнала, який утворюється після підсумування двох ступінчастих сигналів, у командному вікні Matlab введемо послідовність операторів:

```
plot(tout,yout(:,1)), grid
title('Результат підсумовування двох ступінчастих сигналів')
xlabel('t')
```

В результаті у графічному вікні одержимо графік, поданий на рис. 2.10.

Щоб отримати графік сигналу (рис. 2.11), який утворюється після додавання до попереднього сигналу синусоїдального сигналу у командному вікні Matlab введемо послідовність операторів:

```
plot(tout,yout(:,2)), grid  
title('Результат підсумовування двох ступінчастих і синусоїдального  
сигналів')  
xlabel('t')
```

Щоб отримати графік сигналу (рис. 2.12), який утворюється після множення ступінчастого сигналу на сигнал синусоїдальної форми у командному вікні Matlab введемо послідовність операторів:

```
plot(tout,yout(:,3)), grid  
title('Результат множення сигналів')  
xlabel('t')
```

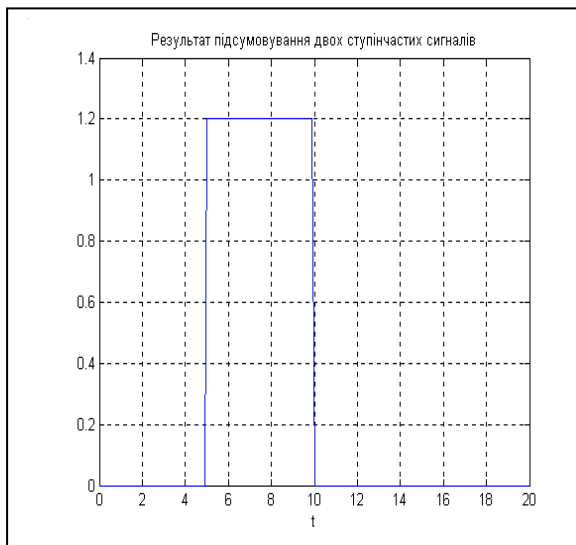


Рис. 2.10. Графік сигналу,
одержаного внаслідок
підсумовування двох
ступінчастих сигналів

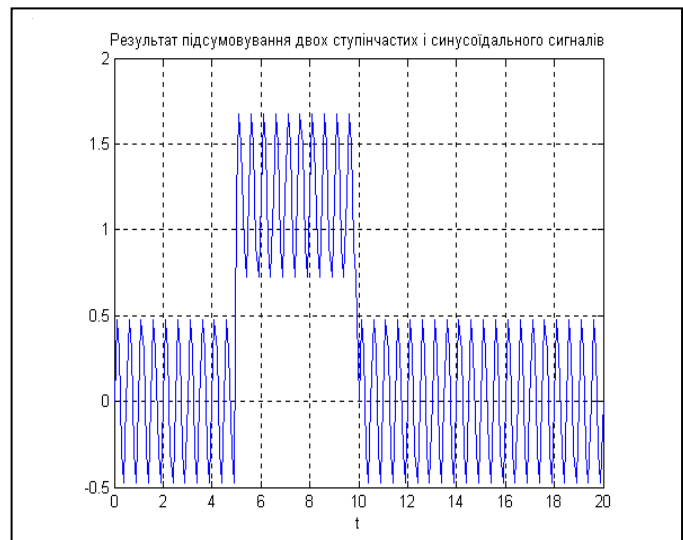


Рис. 2.11. Графік сигналу,
одержаного внаслідок
підсумовування ступінчастого і
синусоїдального сигналів

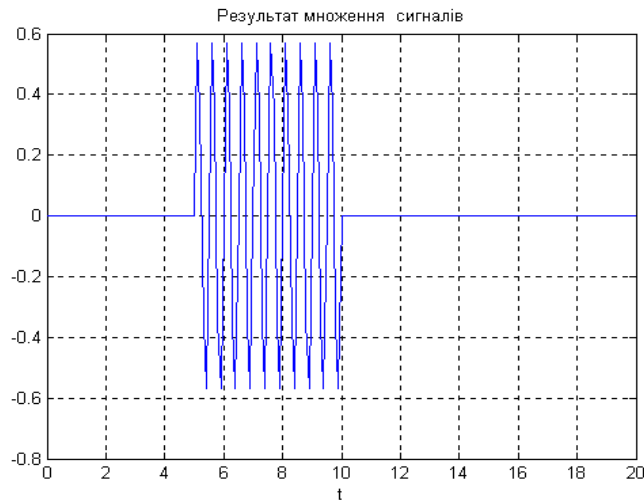


Рис. 2.12. Графік сигнала, одержаного внаслідок множення ступінчастого і синусоїдального сигналів

Отже, у такий спосіб можна одержати різнорівневі сигнали, а також поодинокі пакети імпульсів довільної форми.

Рекомендації до виконання пункту 6. Генерування випадкових процесів

При моделюванні поведінки технічних систем часто виникає потреба використовувати джерела сигналів у вигляді випадкових процесів з заданими характеристиками і з заданими спектральними характеристиками. У Simulink передбачені три блоки генерування випадкових процесів - блок **Random Number**, **Uniform Random Number** і **Band-Limited White Noise**. Щоб сформувати на їхній основі випадкових процес із заданою спектральною щільністю, потрібно "пропустити" сгенерований цими блоками випадковий сигнал через динамічну ланку з певною передатною функцією, яка відповідає потрібній спектральній щільності.

Наприклад, для того, щоб сформувати випадковий процес з деякою переважною частотою λ_m та коефіцієнтом нерегулярності m_i , відповідна

ланка (формулюючи фільтр) повинен мати передатну функцію коливальної ланки такого вигляду:

$$W(s) = \frac{1}{\frac{1}{\text{lam}^2} s^2 + \frac{2 * \text{mu}}{\text{lam}} s + 1}.$$

Такий фільтр моделюється за допомогою блока **Transfer Fcn** (розділ **Continuous**) . Параметри настроювання цього блока дивись на рис. 2.13, а блок-схему формування випадкових сигналів - на рис. 2.14.

Block Parameters: Transfer Fcn

Transfer Fcn	
Matrix expression for numerator, vector expression for denominator. Output width equals the number of rows in the numerator. Coefficients are for descending powers of s.	
Параметры	
Numerator:	[1]
Denominator:	[1/lam^2 2*mu/lam 1]
Absolute tolerance:	auto

Рис. 2.13. Вікно настроювання блока Transfer Fcn

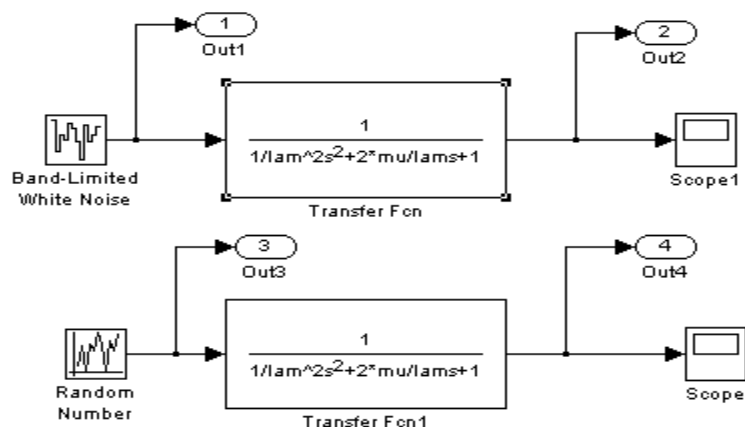


Рис. 2.14. Блок-схема формування випадкових сигналів

Блок-схема формування випадкових сигналів (рис. 2.14) складається з двох стандартних блоків **Random Number** і **Band-Limited White Noise**, однакових формуючих фільтрів на основі блоків **Transfer Fcn**, оглядових блоків **Scope** та чотирьох вихідних портів **Out**. На рис. 2.15 наведені вікна налаштування генераторів випадкових сигналів.

Block Parameters: White Noise.(mask)(link)

- Continuous White Noise. (mask) (link) —————

White noise for continuous (s-domain) systems. Band-limited using zero-order-hold.

- Параметры —————

Noise power:

[0.1]

Sample time:

0.1

Seed:

[23341]

☒ Interpret vector parameters as 1-D

Block Parameters: Random Number

- Random Number —————

Output a normally (Gaussian) distributed random signal. Output is repeatable for a given seed.

- Параметры —————

Mean:

0

Variance:

1

Initial seed:

0

Sample time:

0

☒ Interpret vector parameters as 1-D

Рис. 2.15. Вікна налаштування блоків генерування випадкових сигналів

Після запуску на моделювання, та виконання наступних операторів одержимо графіки випадкових процесів, показані на рис. 2.16 і 2.17:

```
subplot(2,1,1)
plot(tout,yout(:,1)), grid
title('Випадковий процес з Band-limited White Noise')
xlabel('t')
subplot(2,1,2)
plot(tout,yout(:,2)), grid
title('Випадковий процес з переважною частотою lam = 5')
xlabel('t')
subplot(2,1,1)
plot(tout,yout(:,3)), grid
title('Випадковий процес з Random Number')
xlabel('t')
subplot(2,1,2)
plot(tout,yout(:,4)), grid
title('Випадковий процес з переважною частотою lam = 5')
xlabel('t')
```

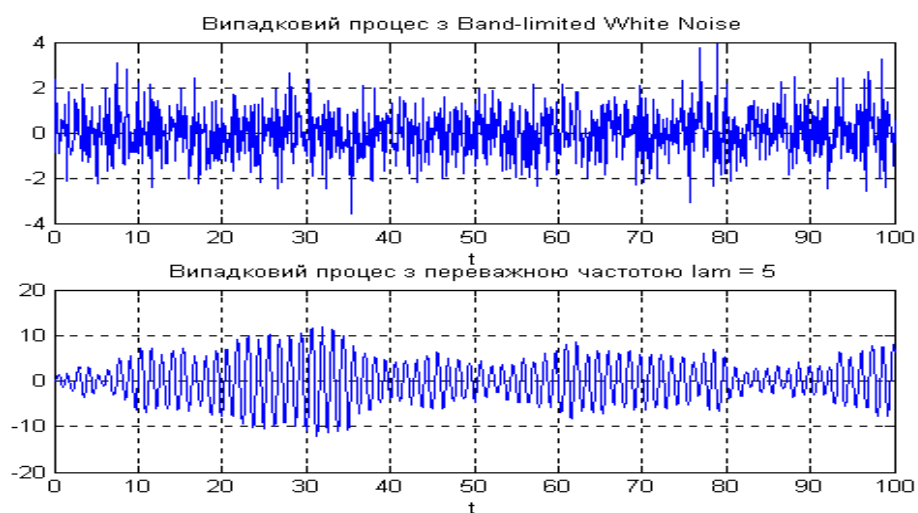


Рис. 2.16. Випадкові процеси з блока Band-Limited White Noise

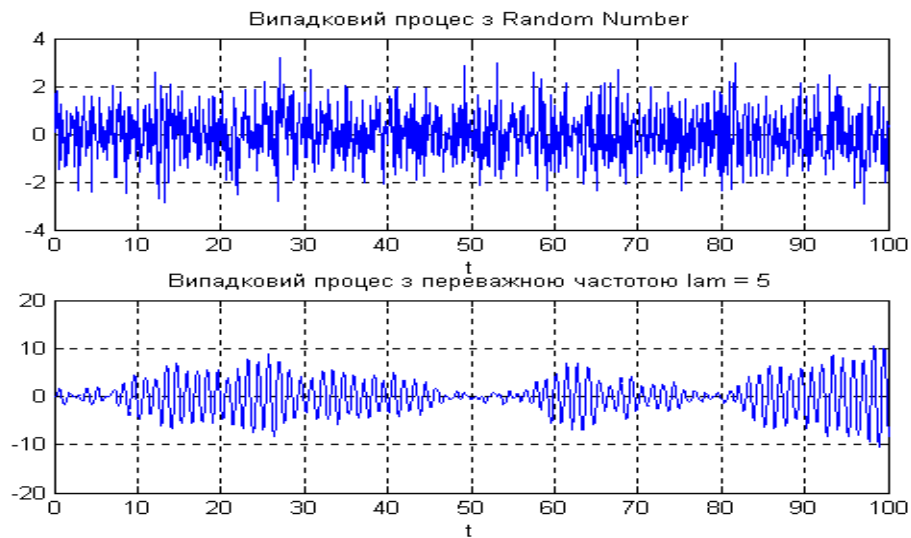


Рис. 2.17. Випадкові процеси з блока Random Number

Отже, у такий спосіб дійсно можна утворити випадкові процеси з заданими спектральними характеристиками.

Рекомендації до виконання пункту 5. Робота з блоком XYGraph.

Для отримання $f(a)$ числового значення функції при $u=a$ використовується блок Fcn (розділ Functions&Tables), на вхід якого подаємо постійну величину a , в вихід якого приєднуємо до блоку Display.

У вікно XYGraph необхідно вивести три графіки: $y=f(u)$, $y=f(a)$, $u=a$, див. рис. 2.18.

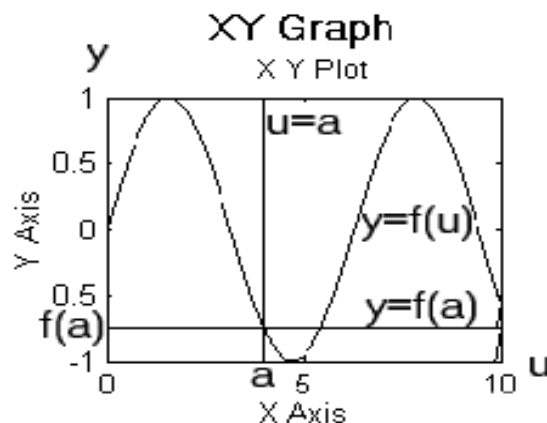


Рис. 2.18. Вікно блоку XYGraph.

Два вхідні сигнали блоку XYGraph повинні змінюватися, згідно таблиці:

Поточний час моделювання	Вхідні сигнали блоку XYGraph		Вихідний сигнал багатопортового перемикача Multiport Switch
	Аргумент вікна XYGraph	Значення функції аргументу вікна XYGraph	
від 0 до 10	U	$f(u)$	1
від 10 до 20	від 10 до 0	$f(a)$ горизонтальна лінія, що виводиться зправа наліво	2
від 20 до 30	A	від -2 до 2 вертикальна лінія, що виводиться вниз вгору	3

Вхідні сигнали блоку XYGraph для поточного часу моделювання від 0 до 10 створюються за допомогою блоків Clock – u та Fcn - $f(u)$.

Вхідні сигнали блоку XYGraph $f(a)$ та a можна взяти із блок-моделі обчислення і виводу на Display числового значення $f(a)$, що була побудована раніше.

Вхідні сигнали блоку XYGraph, що змінюються від 10 до 0 при зміні поточного часу моделювання від 10 до 20, а також що змінюються від -2 до 2 при зміні поточного часу моделювання від 20 до 30 можна створити за допомогою двох блоків Repeating Sequence (розділ Sources).

Вхідні сигнали блоку XYGraph за допомогою трьох блоків Mux (розділ Signal&Systems) потрібно об'єднати у три вектори, відповідно для трьох відрізків поточного часу моделювання.

Для відображення трьох векторів одним блоком XYGraph потрібно організувати їх почергове за часом надходження на вхід XYGraph. Це

можна зробити за допомогою багатопортового перемикача Multiport Switch (розділ Nonlinear), якщо у якості інформаційних сигналів подати на нього три одержані раніше вектори. У якості керуючого сигналу потрібно за допомогою Repeating Sequence (розділ Sources) створити сигнал, що, першу третину часу моделювання дорівнює 1, другу -2, третю -3.

Для подачі кожного з трьох векторів на вхід XYGraph його необхідно розділити на окремі сигнали за допомогою блоку Demux (розділ Signal & Systems). Також необхідно виставити оптимальні параметри блоку XYGraph.

Контрольні запитання

1. Поясніть призначення блоків SineWave, Random Number, Step, Ramp, Clock, Signal Generator, Repeating Sequence, Pulse Generator, ChirpSignal, From Workspace, To Workspace які параметри налаштування блоків, як ці параметри впливають на роботу блоків?
2. Що являє собою робоча область MATLAB?
3. Як побудувати графік функції заданої таблицею значень аргументу та функції у текстовому файлі MS-DOS?
4. Які командні рядки MATLAB використовуються для введення в Workspace значень часу t і вихідного сигналу $f(x)$ у вигляді однієї та двох змінних?
5. Який вигляд мають змінні робочої області MATLAB Workspace і як вони використовуються?
6. Які формати збереження даних у робочу область MATLAB Workspace Ви знаєте?
7. Які блоки використовуються для об'єднання сигналів у вектор, підсумовування і множення сигналів?
8. Які блоки використовуються для генерування випадкових процесів?

9. Поясніть роботу блоку XYGraph, які параметри налаштування блоку?
Які відмінності між блоками Score та XYGraph
10. Поясніть для чого призначений блок Fcn, які параметри налаштування блоку, як ці параметри впливають на роботу блоку?

ЛАБОРАТОРНА РОБОТА №3

МОДЕЛЮВАННЯ МЕХАНІЧНИХ ПРИСТРОЇВ З СУТТЄВИМИ НЕЛІНІЙНОСТЯМИ

Мета роботи: Ознайомитись з роботою інтегруючого блоку, призначення його входів та виходів, впливом значень параметрів інтегруючого блоку на його роботу. Навчитися використовувати інтегруючий блок для моделювання динамічних процесів.

Завдання для лабораторної роботи

1. Створити блок-модель, яка зображена на рис. 3.1 і встановити параметри блоків відповідно варіанту (див. табл. 1). Надрукувати й пояснити графіки, одержані в результаті моделювання у вікні Score.

Варіанти завдання до лабораторної роботи №3

Варіант	Початковий напрямок руху м'яча	Еластичність м'яча	Початкова швидкість м'яча	Початкова висота	Час моделювання	Параметри вікон Score для виводу графіків зміни			
						висоти		швидкості	
						Y _{min}	Y _{max}	Y _{min}	Y _{max}
1	вгору	-0,5	40	10	20	0	100	-15	40
2	униз	-0,8	-35	20	33	0	60	-50	40

Варіанти завдання до лабораторної роботи №3

Варіант	Початковий напрямок руху м'яча	Еластичність м'яча	Початкова швидкість м'яча	Початкова висота	Час моделювання	Параметра вікон Score для виводу графіків зміни			
						висоти		швидкості	
						Y_{min}	Y_{max}	Y_{min}	Y_{max}
3	униз	-0,5	-40	30	40	-15	40	-50	30
4	вгору	-0,6	-60	15	15	0	150	-10	40
5	униз	-0,9	-60	10	20	0	50	-10	50
6	вгору	-0,8	50	20	30	0	100	-50	50
7	вгору	-0,95	40	25	27	-10	100	-15	55
8	униз	-0.5	30	10	16	0	50	-15	20
9	вгору	-0.7	-30	20	20	-10	40	-20	50
10	униз	-0.65	70	40	30	-20	90	-25	60
11	вгору	-0.85	80	30	30	-30	100	-30	70
12	униз	-0.736	-29	50	40	-10	50	-50	40
13	вгору	-0.493	53.6	15	35	-15	40	-20	40
14	вгору	-0.633	-10	8	20	-10	30	-15	30
15	униз	-0.906	-70	43	30	-50	50	-20	40

2. Змінити час моделювання блок-моделі1
3. Зупинити процес моделювання при досягненні м'ячем половини початкової висоти.
4. Змінити блок-модель1 таким чином, щоб у початковий момент часу м'яч рухався в протилежному напрямку, ніж у попередній моделі.
5. Змінюючи блок-модель1, створити блок-модель2 у якій скиданням інтегруючого блоку **Integrator1** керував би сигнал створюваний блоком **Pusle generator** (розділ **Source**). Роздрукувати блок-модель2. Роздрукувати

й проаналізувати сигнали «швидкість», «висота», вихідний сигнал **Pulse generator** і сигнал порту насичення **Integrator2** при всіх можливих значень параметра **External reset** (зовнішнє скидання) блоку **Integrator1**. Пояснити отримані графіки.

6. Створити блок-модель3, яка б моделювала падіння непружного каменю з певної висоти з нульовою початковою швидкістю. Початкову швидкість задати як власну установку початкового значення блоку **Integrator1**. Початкову висоту задавати ззовні блоку **Integrator2** від блоку констант. Надрукувати блок-модель3. Надрукувати й пояснити зміни в часі сигналу швидкості й висоти падіння каменя.

7. Змінюючи блок-модель1, створити блок-модель4, яка б відрізнялася від вихідної знаками величин, тобто, прискорення додатня, висота – від’ємна, а знак швидкості залежить від напрямку руху м’яча. Тобто результати роботи блоку-моделі4 повинні бути дзеркальним відображенням результатів роботи блоку-моделі1, тобто змінити напрямки осей ординат системи координат.

Звіт по лабораторній роботі повинен містити блоків-моделі 2,3,4. Параметри блоків **Integrator1** й **Integrator2** всіх блок-моделей. Графіки з вікна Score для всіх блок-моделей. Всі лінії зв’язку блок-моделей повинні супроводжуватись назвами, що вказують, які сигнали по них проходять.

Робота блок-моделі руху стрибаючого м’яча

Блок-модель див. рис. 3.1 моделює рух стрибаючого м’яча, що кинутий з висоти 10 м униз із початковою швидкістю 50 м/с.

М’яч буде рухатися з постійним прискорення вільного падіння тому сигнал прискорення можна моделювати за допомогою блоку **Constant** (константа) – джерело постійної величини ($-9,8 \text{ м/с}^2$). Значення $(-9,8)$ задається як параметр блоку **Constant**. Знак “-” для даної моделі мають вектори спрямовані вниз, тобто при русі м’яча вниз швидкість буде мати

від'ємні значення, а нагору - додатні. Висота м'яча в такій системі координат буде виражатися додатніми числами.

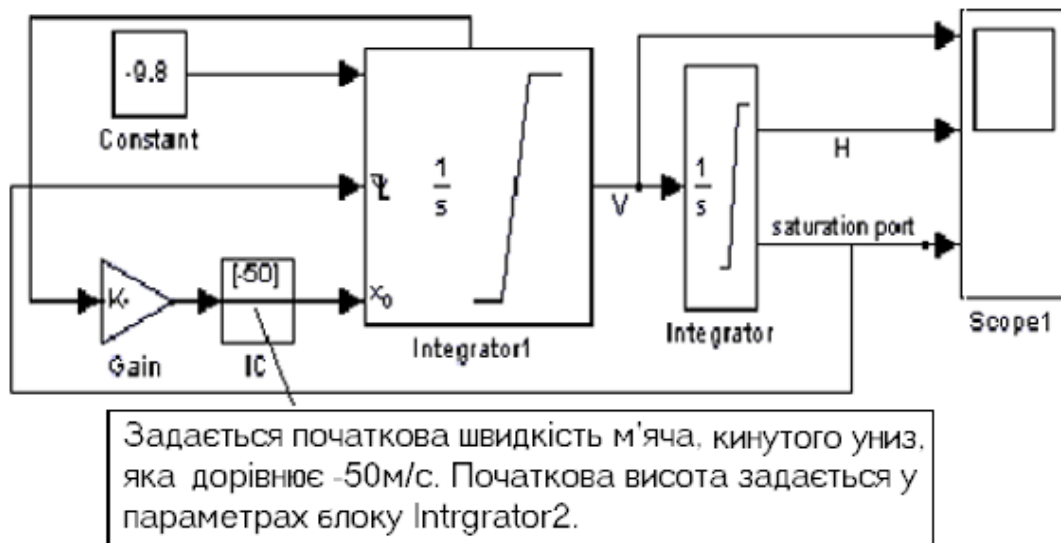


Рис. 3.1. Блок-модель, що моделює рух стрибаючого м'яча

Прискорення вільного падіння подається на вхід блоку **Integrator1** (інтегратор), на виході якого отримаємо швидкість V , бо швидкість – це інтеграл за часом від прискорення:

$$V = \int_{t_1}^{t_2} g dt$$

Отриманий на виході блоку **Integrator1** сигнал, що представляє собою швидкість, подається на вхід ще одного інтегруючого блоку **Integrator2**, на виході якого отримуємо переміщення H , бо переміщення – це інтеграл за часом від швидкості:

$$H = \int_{t_1}^{t_2} V dt$$

Для перегляду графіків зміни у часі швидкості й переміщення ці сигнали подаються на вхід блоку **Scope** (індикатор), результати роботи якого зображені на рис. 3.4.

Завдання початкових значень висоти й швидкості в інтегруючих блоках. У початковий момент моделювання м'яч має початкову швидкість і перебуває на певній висоті. Ці початкові умови необхідно врахувати в процесі моделювання.

Початкова швидкість м'яча (-50 м/с - рух униз) подається від блоку **IC (Initial Condition** – початковий стан). Цей блок дозволяє встановити початковий стан сигналу, причому тривалість перебування системи у відповідному стані дорівнює тривалості кроку моделювання, а потім блок пропускає вхідний сигнал на вихід без зміни. Таким чином, за допомогою блоку IC початкова швидкість подається тільки в початковий момент моделювання на відміну від прискорення (-9.81), що подається увесь час, поки працює модель.

Початкова швидкість подається на спеціальний вхід інтегруючого блоку **Integrator1**, призначений для установки початкового значення інтегратора при скиданні, див. рис. 3.2 і рис. 3.3. Далі в процесі моделювання на цей же вхід інтегратора буде надходити кожне нове значення швидкості м'яча, яку він набуватиме після відскоку від землі за рахунок його пружності.

Вхід інтегратора для установки початкового значення може бути включений або виключений. Для цього існує параметр блоку **Initial condition Source** (джерело початкового стану). Якщо значення цього параметра має значення `external` (ззовні), то вхід буде включений і на нього від інших блоків повинен надходити сигнал з початковим значенням, як у блоці **Integrator1**. Якщо значення параметра `intrnal`, то початкове значення не надходить ззовні, а задається в самому інтегруючому блоці значенням його параметра **Initial Conditional**.

Block Parameters: Integrator 1

Integrator
Continuous-time integration of the input signal.

Parameters

External reset: falling вхід скидання блоку включений, причому скидання відбувається при керуючому сигналі, який зменшується

Initial condition source: external початкове значення блоку задається із зовнішнього блоку

☐ **Limit output**

Upper saturation limit:
inf

Lower saturation limit:
-inf

☐ **Show saturation port** включений порт стану

☒ **Show state port** точність обчислень

Absolute tolerance:
auto

Рис. 3.2. Параметри інтегруючого блоку Integrator 1.

Block Parameters: Integrator 2

Integrator
Continuous-time integration of the input signal.

Parameters

External reset: none вхід скидання блоку виключений

Initial condition source: Internal початкове значення блоку задається, як параметр самого блоку у полі Initial condition

Initial condition:
10 початкова висота м'яча

☒ **Limit output** включені обмеження вихідного значення, які впливають на формування сигналу в порту насичення

Upper saturation limit:
inf

Lower saturation limit:
0 верхнє граничне значення не обмежено

☒ **Show saturation port** нижнє граничне значення - висота 0

☐ **Show state port** включений порт насичення

Absolute tolerance:
auto

Рис. 3.3. Параметри інтегруючого блоку Integrator2.

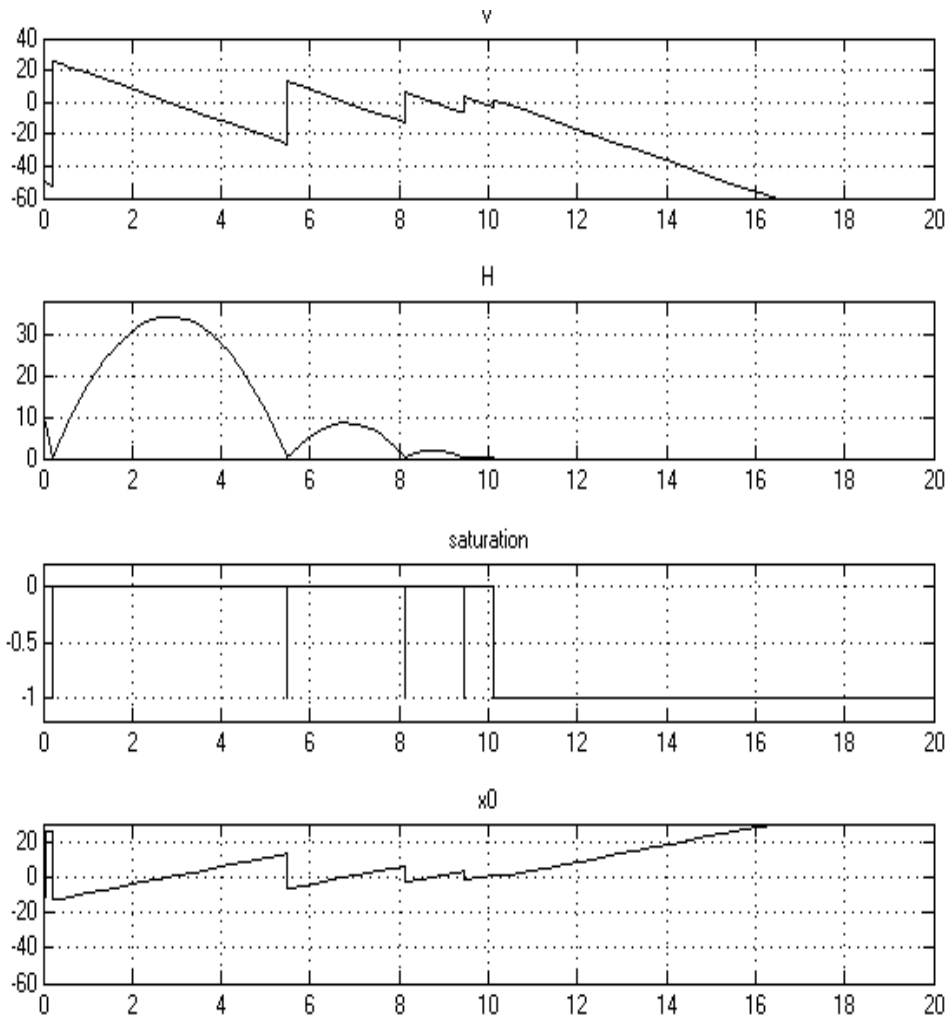


Рис. 3.4. Вікно блоку Score із графіками зміни за часом швидкості – V, висоти – H стрибаючого м'яча, вихідного сигналу порту насичення блоку Integrator2– saturation, початкового значення блоку Integrator1 – x0

В інтегруючому блоці **Integrator2** вхід початкового значення виключений, а початкова висота м'яча 10м задається значенням параметра **Initial Conditional** (початковий стан) блоку **Integrator2**.

Таким чином, у блоці **Integrator1** початкове значення (-50) встановлюється ззовні від блоку **IC**. І тому значення **Initial Conditional source** встановлено **external** (зовнішня), а в блоці **Integrator2** використовується власна установка початкового значення й тому в полі

Initial Conditional source установлене **internal**, а в полі **Initial Condition** значення початкової висоти (10).

Скидання інтегруючого блоку **Integrator1** необхідно робити щораз при падінні м'яча на землю ($H=0$). Після скидання початкове значення блоку **Integrator1**, а саме: швидкість буде дорівнює добуткові $(-K*V)$, де числове значення K визначається пружністю м'яча, а зміна знака обумовлена зміною напрямку руху м'яча: униз – до скидання й нагору – після.

Скидання інтегруючого блоку відбувається при надходженні сигналу в його спеціальний порт, який називається вхід скидання. Цей вхід може бути включений вибором значення в полі **External reset** (зовнішнє скидання) параметрів блоку. Причому скидання буде відбуватися тільки при надходженні саме спадаючого керуючого сигналу, тому що в полі **External reset** задане значення **falling** (спад).

Після надходження сигналу скидання від **Integrator2** у блоці **Integrator1** відбувається наступне: на спеціальному виході **Integrator1**, що називається вихід стану, генерується сигнал. Виникнення цього сигналу приводить до того, що поточне значення швидкості (кінцева швидкість у момент падіння м'яча на землю) перемножується в блоці **Gain** (підсилювач) на коефіцієнт $(-K)$. Отриманий добуток подається на вхід початкового значення **Integrator1**, а також на блок **Scope**. Далі відбувається процес моделювання вже з новою $(-K)*V$ початковою швидкістю м'яча.

Вихід стану може бути виключений, а включається полем **Show state port** (показати порт стану).

Формування сигналу скидання в блоці **Integrator2**. Сигнал скидання формується блоком **Integrator2** у моменти, коли м'яч черговий раз падає на землю. Цей сигнал використовується для скидання блоку

Integrator1. Скидання **Integrator1** необхідне для установки в цьому блоці нового початкового значення швидкості м'яча. Нове початкове значення швидкості м'яча необхідно задавати тому що після падіння на землю м'яч відскакує нагору зі швидкістю, величина якої залежить від пружності м'яча.

Сигнал скидання генерується на спеціальному виході насичення інтегруючого блоку **Integrator2**. Вихід (порт) насичення може включатися/вимикатися заданням значення поля **Show saturation port** (показати порт насичення) параметрів блоку.

Вихідний сигнал порту насичення **Integrator2** подається на вхід скидання блоку **Integrator1** і управляє роботою блоку **Integrator1**, а також у блок **Scope** (індикатор) рис. 3.3 для перегляду графіка.

Умова формування керуючого сигналу скидання задається в параметрах блоку **Integrator2**, а саме: прапорець **Limit output** (обмеження вихідного значення) встановлене; **Upper saturation limit** (верхнє граничне значення) не обмежене – значення **Inf**; **Lower saturation limit** (нижнє граничне значення) – висота 0. Таким чином, коли висота $H=0$ у порту насичення **Integrator2** формується сигнал, що надходить на вхід скидання **Integrator1**.

Контрольні запитання

1. Поясніть призначення блоку Integrator.
2. Для чого використовується вхід скидання блоку Integrator, які параметри блоку задають режими роботи входу скидання?
3. Для чого використовується вхід початкового значення блоку Integrator, які параметри блоку задають режими роботи входу початкового значення?

4. Для чого використовується вихід насичення блоку Integrator, які параметри блоку задають режими роботи виходу насичення?
5. Для чого використовується вихід стану блоку Integrator, які параметри блоку задають режими роботи виходу стану?
6. Як змінити час моделювання?
7. Який блок зупиняє процес моделювання, які його параметри налаштування?
8. Як у блок-моделі задається початкова швидкість та початкова висота м'яча?
9. Як у блок-моделі задається прискорення вільного падіння?
10. Який блок використовується для перегляду графіків зміни швидкості й переміщення?
11. Поясніть призначення та роботу блоку Initial Condition.

ЛАБОРАТОРНА РОБОТА №4

МОДЕЛЮВАННЯ ІЗ ЗАСТОСУВАННЯМ ОПЕРАЦІЙ ДИФЕРЕНЦІЮВАННЯ ТА ІНТЕГРУВАННЯ

Мета роботи: Навчитися задавати функцію за допомогою блоку Fcn і зростаючий аргумент функції за допомогою блоку Clock. Навчитися визначати приріст числового значення сигналу за певний проміжок часу, а також підсумовувати та зберігати данні, що зодовільняють певній умові на інтервалі моделювання за допомогою блоків Data Store Memory, Data Store Read, Data Store Write, що використовуються для організації спільної області пам'яті та обміну між підсистемами однієї системи. Ознайомитись з роботою блоків аналогового диференціювання Derivate, блоку Quantizer.

Завдання до лабораторної роботи

1. Створити блок-модель, що диференціює функцію $f_1(u)$ з використанням блоку аналогового диференціювання **Derivate**. Графіки функції й диференціала функції вивести у вікно Score в одній системі координат. Рекомендації до виконання всіх пунктів завдання дивись далі.

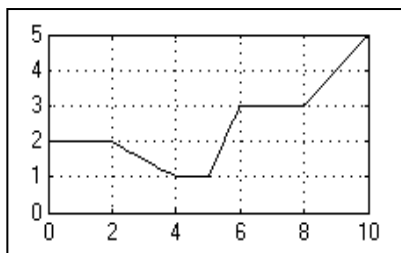
Варіанти завдань до пункту завдання 1.

№ варіанту	Функція $f_1(u)$	№ варіанту	Функція $f_1(u)$
1	$\sin(5u)$	9	$2\sin(3u)$
2	$\sin(3u)$	10	$\cos(u)+1$
3	$\cos(u+1)$	11	$\sin(2u)+2$
4	$2\cos(u+1)-1$	12	$3\cos(2u)-1$
5	$\operatorname{tg}(u)$	13	$1/\cos(u)$
6	$\operatorname{ctg}(u)$	14	$1/\sin(u+1)$
7	$\operatorname{tg}(u+1)$	15	$(1/\sin(u))-1$
8	$\operatorname{tg}(u+1)+1$		

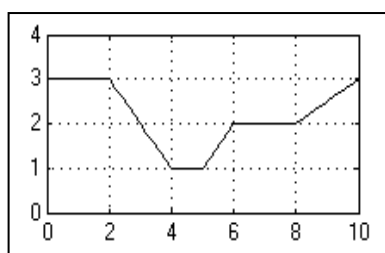
2. Створити блок-модель, що диференціює функцію $f_1(u)$ без використання блоку Derivate. Диференціал визначати як відношення приросту функції до приросту аргументу.
3. Створити блок-модель для знаходження інтеграла функції $f_2(u)$ з використанням блоку аналогового інтегратора **Integrator**. Значення інтеграла за заданий час інтегрування вивести в блок **Display** (розділ **Sinks**). Графіки функції й інтеграла функції вивести у вікно Score в одній системі координат.

Варіанти завдань до пункту завдання 3.

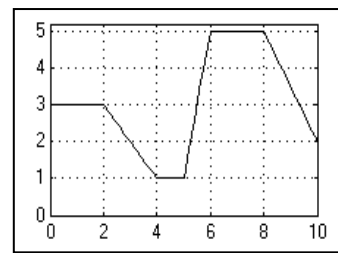
Варіант 1. $f_2(u)$:



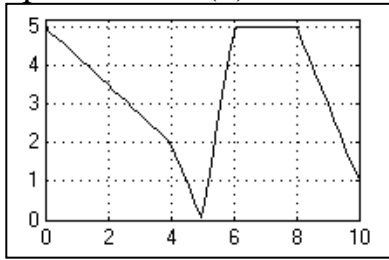
Варіант 2. $f_2(u)$:



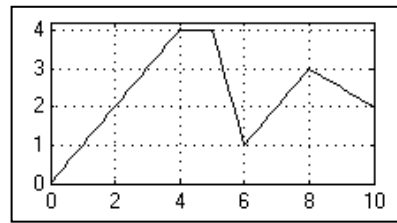
Варіант 3. $f_2(u)$:



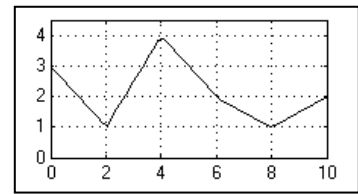
Варіант 4. $f_2(u)$:



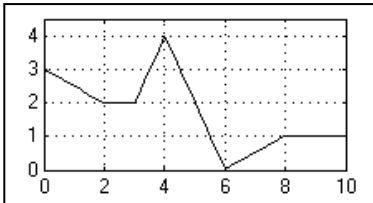
Варіант 5. $f_2(u)$:



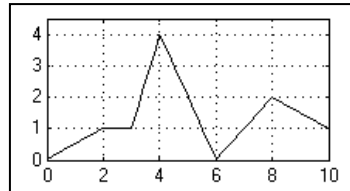
Варіант 6. $f_2(u)$:



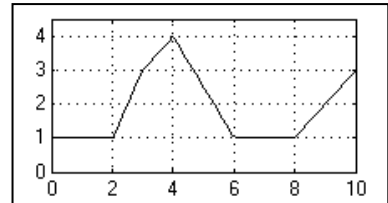
Варіант 7. $f_2(u)$:



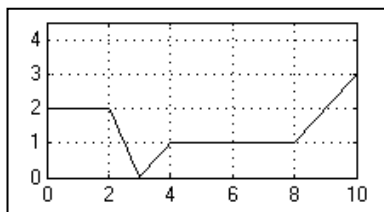
Варіант 8. $f_2(u)$:



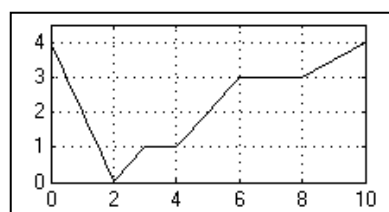
Варіант 9. $f_2(u)$:



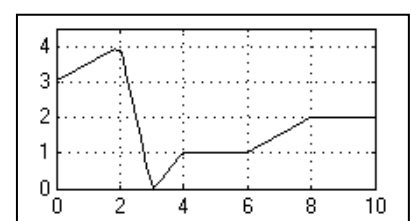
Варіант 10. $f_2(u)$:



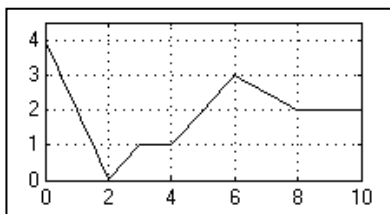
Варіант 11. $f_2(u)$:



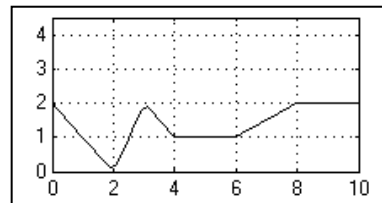
Варіант 12. $f_2(u)$:



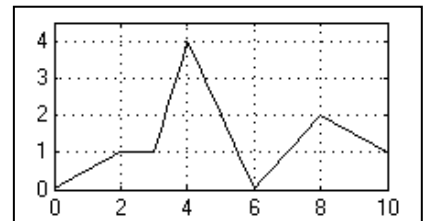
Варіант 13. $f_2(u)$:



Варіант 14. $f_2(u)$:



Варіант 15. $f_2(u)$:



4. Створити блок-модель для знаходження інтеграла функції $f_2(u)$ без використання блоку Integrator. Інтеграл визначити як площу фігури, що знаходиться під функцією.
5. Створити блок-модель для знаходження інтеграла функції $f_1(u)$ згідно варіанта використовуючи блок Integrator, і як площу фігури, що знаходиться під графіком функції. Для обчислення площі фігури, функцію попередньо дискретизувати за допомогою блоку Quantizer (розділ Nonlinear). В одних координатних осях вікна Scope вивести графіки функції, дискретизованої функції, інтеграла функції.

Звіт по лабораторній роботі повинен містити:

1. П'ять блок-моделей диференціювання й інтегрування функцій з використанням відповідних блоків і без використання.
2. Графіки функції $f_1(u)$ й диференціала функції в одній системі координат (результат виконання пункту 1 завдання).
3. Графіки функції $f_2(u)$ й інтеграла функції в одній системі координат (результат виконання пункту 3 завдання).
4. Формулу для визначення інтеграла функції $f_2(u)$ за час t (для пункту 4 завдання).
5. Вікно Scope с графіками функції $f_1(u)$ до й після дискретизації, інтеграла функції $f_1(u)$ (результат виконання пункту 5 завдання).
6. Дослідити залежність помилки обчислення інтеграла функції від інтервалу квантування блоку Quantizer і максимального кроку моделювання Simulation Parameter. Дослідити залежність помилки обчислення інтеграла дискретної функції від інтервалу квантування, а також від параметрів моделювання Simulation Parameter, а саме від величин максимального кроку моделювання, відносної похибки моделювання (Relative tolerance), обраного методу моделювання.

Рекомендації для виконання пункту 1 завдання.

- Функцію задати за допомогою блоку **Fcn** (розділ Function&Table) і **Clock** (розділ Sources) для завдання зростаючого аргументу функції.
- Для диференціювання функції використати блок **Derivate** (розділ Continuous).
- Функцію й диференціал вивести в блок **Scope**, об'єднавши попередньо їх значення у вектор, за допомогою блоку **Mux** (розділ Signal&System).

Рекомендації для виконання пункту 2 завдання. Диференціал функції це відношення приросту функції до приросту аргументу:

$$Y'(u) = \frac{dY}{du}$$

- Приріст функції визначити, як різницю поточного й попереднього значень. Попереднє значення функції можна отримати за допомогою блоку **Memory** (розділ Continuous), а різницю за допомогою блоку **Sum** (розділ Math) див. рис. 4.5.
- Приріст аргументу визначається аналогічно визначенню приросту функції.
- Прирост функції діляться на прирост аргументу за допомогою блоку **Math Function** (розділ **Math**, функція reciprocal).
- Функцію й диференціал вивести у вікні **Scope** і порівняти з результатами, отриманими при виконанні пункту 1 завдання.

Рекомендації для виконання пункту 3 завдання.

- Значення функції задати за допомогою блоку **Repeating Sequence** (розділ Sources).
- Проінтегрувати функцію, використовуючи блок **Integrator** (розділ Continuous).
- Вивести результати у вікно **Scope** і на **Display** (розділ Sinks).

Рекомендації для виконання пункту 4 завдання. Інтеграл функції за час t (якщо час є аргументом функції) визначається як площа, обмежена осями OY , OX , прямий $Y=t$ і графіком функції (див. рис. 4.1). Площу знаходимо як суму площ прямокутників і трикутників для кожного інтервалу зміни аргументу, а саме:

$$A = A + S_{\text{одного інтервалу}},$$

де A – загальна площа;

$$S_{\text{одного інтервалу}} = S_{\text{прямокутника}} + S_{\text{трикутника}} = Y_{\min} \cdot dT + \frac{(Y_{\max} - Y_{\min}) \cdot dT}{2}$$

При цьому на ділянках графіка, де площу трикутника додавати не потрібно, значення площі трикутника дорівнює 0.

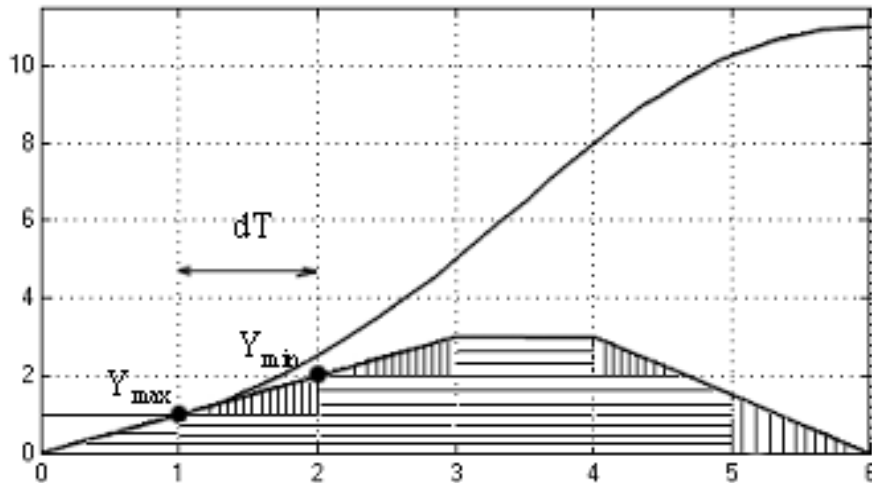


Рис. 4.1. Графік функції й інтеграла функції.

- Величину інтервалу часу (приріст аргументу) dT визначити так само, як у пункті 2.
- Попереднє значення функції визначити за допомогою блоку **Memory** (розділ Continuous).
- Обидва значення функції попереднє й поточне відіслати у два блоки **MinMax** (розділ **Math**) один із яких визначить Y_{\min} , а другий – Y_{\max} .
- Y_{\min} і величину інтервалу часу dT відсилаємо в блок **Product** (розділ **Math**) для визначення $S_{\text{прямокутника}}$ для інтервалу dT
- За допомогою блоку **Sum** (розділ **Math**) визначити різницю $Y_{\max} - Y_{\min}$, яку множимо на dT і ділимо на 2 для визначення $S_{\text{трикутника}}$ для інтервалу dT .
- Площі прямокутників і трикутників підсумувати, використовуючи блоки **Data Store Memory**, **Data Store Read**, **Data Store Write** (розділ Signal& System). Приклад блок-діаграми, яка підсумовує інтервали часу наведено далі.

- Порівняти значення інтеграла функції за час t отримане за допомогою блоку Integrator й як площу фігури.

Рекомендації для виконання пункту 5 завдання. Для того, щоб знайти площу фігури, що знаходиться під графіком функції, яка змінюється плавно, потрібно здійснити дискретизацію функції за допомогою блоку Quantizer (розділ Nonlinear) див. рис. 4.2. У цьому випадку точність обчислення інтеграла буде істотно залежати від інтервалу квантування, що задається в параметрах блоку Quantizer (див. рис. 4.3) і від максимального кроку моделювання, що задається параметрами моделювання (див. рис. 4.4). Блок-схема аналогічна блок-схемі, яка отримана при виконанні пункту 4, а відрізняється лише наявністю блоку Quantizer.

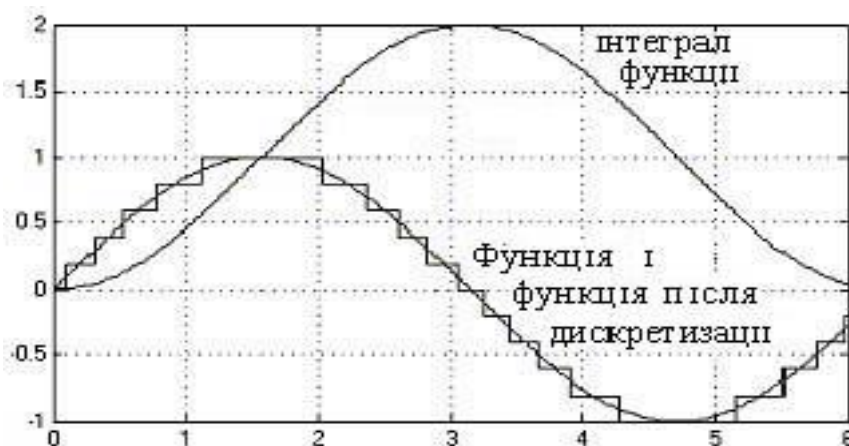


Рис. 4.2. Обчислення інтеграла функції з використанням дискретизації.

Block Parameters: Quantizer

Quantizer

Discretize input at given interval.

Parameters

Quantization interval:

0.1

☒ Treat as gain when linearizing

Рис. 4.3. Параметри блоку Quantizer (розділ Nonlinear).

Simulation Parameters: intdif

Solver | Workspace I/O | Diagnostics | Advanced | Real-Time Workshop

Simulation time

Start time: Stop time:

Solver options

Type:

Max step size: Relative tolerance:

Min step size: Absolute tolerance:

Initial step size:

Output options

Refine factor:

Рис. 4.4. Параметри моделювання Simulation Parameters моделі.

Робота прикладу блок-діаграми використання блоків Data Store Memory, Data Store Read, Data Store Write (див. рис. 4.5 і 4.6).

Блок-діаграма підсумовує інтервали часу. У даному демонстраційному прикладі підсумуються всі інтервали часу, які були присутні протягом усього часу моделювання тому отримана сума (блок **Display**) дорівнює часу моделювання заданому в **Simulation Parameters** у вікні **Stop Time**.

Для одержання інтервалу часу блок **Sum** віднімає від поточного значення часу попереднє. Попереднє значення часу отримується за допомогою блоку **Memory**.

Для підсумовування інтервалів використовується блок **Data Store Memory**, у якому зберігається сума. Блоки **Data Store Read** й **Memory** дозволяють одержати попереднє й значення суми, до якого додається інтервал й отримана нова сума зберігається блоком **Data Store Write**. Таким чином, $A=A+(\text{інтервал часу})$, де A – поточна сума інтервалів часу. Блок **Data Store Read** можна використати для підсумовування не тільки часу, а і будь-яких величин.

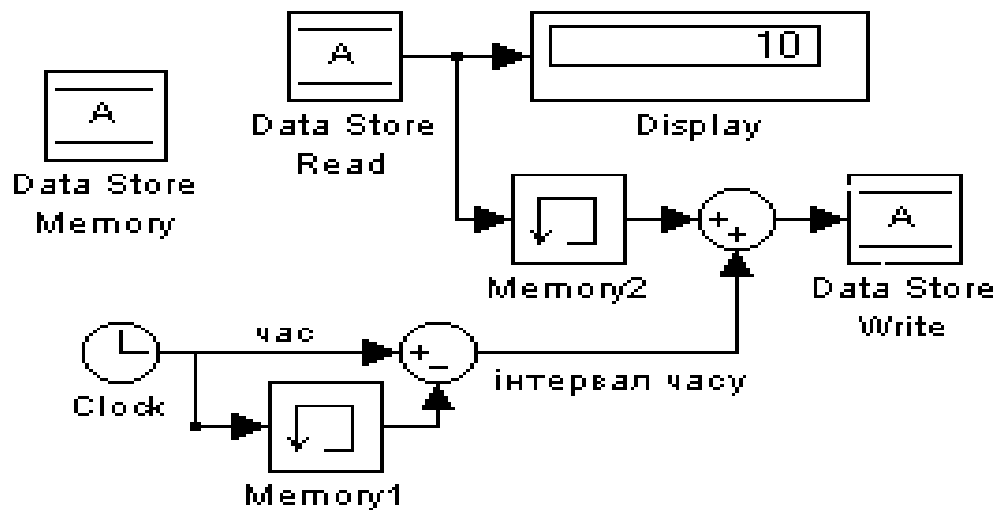


Рис. 4.5. Приклад роботи блоків Data Store Memory, Data Store Read, Data Store Write.

Solver	Workspace I/O	Diagnostics	Advanced	Real-Time Workshop
Simulation time				
Start time:	0	Stop time:	10	
Solver options				
Type:	Variable-step	discrete (no continuous states)		
Max step size:	0.001			
Output options				
Refine output		Refine factor:	1	

Рис. 4.6. Значення параметрів Simulation Parameters для приклада роботи блоків Data Store Memory, Data Store Read, Data Store Write.

Контрольні запитання

1. Поясніть призначення блоку Fcn, які параметри цього блоку?
2. Поясніть призначення блоку Derivate, які параметри цього блоку?
3. Поясніть призначення блоку Integrator, які параметри цього блоку?
4. Поясніть призначення блоку Quantizer, які параметри цього блоку?

5. Поясніть призначення блоку Memory, які параметри цього блоку?
6. Приведіть модель, вихідним сигналом якої є приріст вхідного сигналу за один крок моделювання.
7. Поясніть призначення блоку Product, які параметри цього блоку?
8. Поясніть призначення блоків Data Store Memory, Data Store Read, Data Store Write, які параметри цих блоків?
9. Приведіть модель, вихідним сигналом якої є сума інтервалів часу.

ЛАБОРАТОРНА РОБОТА №5

МОДЕЛЮВАННЯ РОБОТИ ЛОГІЧНИХ СХЕМ

Мета роботи: Ознайомитись з роботою блоків **Relational Operator**, що реалізує операції відношення над двома вхідними сигналами, **Combinatorial Logic** (комбінаторика), що забезпечує перетворення вхідних булевих величин у вихідну згідно таблиці істинності, **IF** для організації розгалуження процесу моделювання в залежності від виконання умови.

Завдання до лабораторної роботи

Скласти блок-діаграму в якій значення функції f1 порівнюється зі значеннями функцій f2, f3, f4, а до результатів порівняння пред'явлені вимоги відповідно з варіантом.

У блок - діаграмі забезпечити:

1. Вивод в одних осях координат вікна **Scope** графіків всіх 4-х функцій, а також сигналу-результату перевірки виконання умов (виводити 1, якщо всі умови виконуються, 0 – не виконується хоча б одне з умов).
2. Визначення сумарного часу, протягом якого виконуються всі 3 умови, і вивести його значення в блок **Display**.

3. Вивод у робочу область **MathLab (Workspace)** моментів часу, у які починалися або навпаки припиняли виконуватися вимоги умов.

Варіанти завдання до лабораторної роботи №5.

№	Функції $f_1(u)$, $f_2(u)$, $f_3(u)$, $f_4(u)$	Вимоги до виконання умов порівняння значень функцій		
1	$f_1(u)=\sin(5u)$ $f_2(u)=\cos(u+1)$	$f_3(u)=2\sin(u)$ $f_4(u)=\sin(u)+3$	$f_1 < f_2$	$f_1 < f_3$ $f_1 < f_4$
2	$f_1(u)=\sin(3u)$ $f_2(u)=\sin(2u+1)$	$f_3(u)=\cos(u+3)$ $f_4(u)=\sin(u)+1$	$f_1 > f_2$	$f_1 > f_3$ $f_1 < f_4$
3	$f_1(u)=\cos(u+2)$ $f_2(u)=\sin(3u)$	$f_3(u)=\cos(3u)$ $f_4(u)=\cos(u+1)$	$f_1 > f_2$	$f_1 < f_3$ $f_1 > f_4$
4	$f_1(u)=\sin(u)$ $f_2(u)=\cos(u+2)$	$f_3(u)=\sin(2u+3)$ $f_4(u)=\sin(3u)$	$f_1 > f_2$	$f_1 > f_3$ $f_1 < f_4$
5	$f_1(u)=\sin(2u+1)$ $f_2(u)=\cos(u+3)$	$f_3(u)=\sin(3u)$ $f_4(u)=\cos(u+1)$	$f_1 < f_2$	$f_1 > f_3$ $f_1 < f_4$
6	$f_1(u)=\cos(u)$ $f_2(u)=\sin(u+3)$	$f_3(u)=\sin(u+2)$ $f_4(u)=\cos(3u)$	$f_1 > f_2$	$f_1 > f_3$ $f_1 > f_4$
7	$f_1(u)=2\cos(u)$ $f_2(u)=3\sin(u)$	$f_3(u)=\cos(2u+1)$ $f_4(u)=\cos(3u-1)$	$f_1 < f_2$	$f_1 > f_3$ $f_1 > f_4$
8	$f_1(u)=\sin(u)/\cos(u)$ $f_2(u)=\sin(2u+3)$	$f_3(u)=\sin(3u)$ $f_4(u)=\sin(u)$	$f_1 > f_2$	$f_1 > f_3$ $f_1 > f_4$
9	$f_1(u)=\cos(u)/\sin(u)$ $f_2(u)=\cos(2u+3)$	$f_3(u)=\cos(3u)$ $f_4(u)=\sin(u)$	$f_1 > f_2$	$f_1 > f_3$ $f_1 > f_4$
10	$f_1(u)=1/\sin(u)$ $f_2(u)=1/\cos(u)$	$f_3(u)=\sin(u+2)$ $f_4(u)=\cos(u+3)$	$f_1 > f_2$	$f_1 > f_3$ $f_1 > f_4$

Варіанти завдання до лабораторної роботи №5.

№	Функції $f_1(u)$, $f_2(u)$, $f_3(u)$, $f_4(u)$	Вимоги до виконання умов порівняння значень функцій		
11	$f_1(u)=\sin(3u)$ $f_2(u)=\sin(2u+1)$	$f_3(u)=\cos(u)/\sin(u)$ $f_4(u)=\cos(u)$	$f_1 < f_2$	$f_1 < f_3$ $f_1 > f_4$
12	$f_1(u)=\cos(u+3)$ $f_2(u)=3\sin(u)$	$f_3(u)=\cos(u)$ $f_4(u)=\sin(u)$	$f_1 < f_2$	$f_1 < f_3$ $f_1 < f_4$
13	$f_1(u)=2\sin(u)$ $f_2(u)=\sin(2u)$	$f_3(u)=\sin(2u-2)$ $f_4(u)=\cos(u-1)$	$f_1 > f_2$	$f_1 > f_3$ $f_1 < f_4$
14	$f_1(u)=\cos(3u-1)$ $f_2(u)=3\cos(u-1)$	$f_3(u)=2\sin(u)$ $f_4(u)=\cos(2u)$	$f_1 < f_2$	$f_1 < f_3$ $f_1 < f_4$
15	$f_1(u)=\sin(u+2)$ $f_2(u)=\sin(2u+3)$	$f_3(u)=\sin(u)$ $f_4(u)=\cos(2u)$	$f_1 > f_2$	$f_1 > f_3$ $f_1 < f_4$

Звіт по лабораторній роботі повинен містити:

1. Блок - діаграму.
2. Вікно **Score** із графіками функцій і сигналом-результатом перевірки умов в одних осях координат.
3. Значення сумарного часу виконання умов.
4. Значення моментів часу початку або навпаки припинення виконання вимог умов з робочої області Workspace.

Рекомендації до виконання пункту 1 завдання.

- За допомогою блоків **Fcn** (розділ Function&table) організувати обчислення значень всіх функцій, використовуючи в якості їхніх аргументів вихідний сигнал блоку **Clock** (розділ Sources).
- Для одержання 4-х графіків в одній системі координат значення функцій об'єднати у вектор за допомогою блоку **Mux** (розділ Signal&System) і отриманий вектор вивести в **Scope** (розділ Sinks).
- За допомогою трьох блоків **Relation Operator** (розділ Math) організувати контроль виконання умов завдання.
- Результати контролю об'єднати у вектор і подати на вхід блоку **Combinatorial Logic** (розділ Math), на виході якого забезпечити 1 – при виконанні всіх умов, або 0 – при невиконанні хоча б однієї з умов. Отриманий сигнал приєднати до вектора значень функцій для передачі у блок Scope.

Рекомендації до виконання пункту 2 завдання. Для визначення сумарного часу виконання 3-х умов визначимо суму добутків інтервалів часу (крок моделювання) на вихідний сигнал **Combinatorial Logic**. Час невиконання умов не увійде в суму тому що він буде множитися на 0.

- Для визначення величини інтервалів часу використати блоки **Memory** (розділ Continuous), **Sum** (розділ Math) і вихідний сигнал (час) блоку **Clock** (розділ Sources).
- Використовуючи блок **Product** (розділ Math), отримане значення інтервалу часу помножити на вихідний сигнал блоку **Combinatorial Logic**.
- Для підсумовування добутків, отриманих на кожному з інтервалів модельного часу, використати блоки **Data Store Memory, Data Store Read, Data Store Write** (розділ Signal&System). Отриману суму вивести за допомогою блоку **Display** (розділ Sinks).

Рекомендації до виконання пункту 3 завдання. Тільки в момент часу, коли починають або перестають виконуватися всі 3 умови сума попереднього й поточного вихідного сигналів блоку **Combinatorial Logic** буде дорівнює 1. З огляду на цей факт, можна використати блок **If** (розділ Subsystem) для виводу в робочу область (Workspace) цих моментів часу. Приклад роботи блоку **If** наведений далі.

- Для визначення суми попереднього й поточного вихідного сигналів блоку **Combinatorial Logic** використаємо блоки **Memory** (розділ Continuous), **Sum** (розділ Math) аналогічно визначенню інтервалу часу.
- Отриману суму використати як керуючий сигнал для блоку **If** (розділ Subsystem). Якщо сума дорівнює 1, то вивести в робочу область значення часу (вихідний сигнал блоку Clock) за допомогою блоку **To Workspace** (розділ Sinks).

Робота блок-діаграми прикладу використання блоку If (рис. 5.1)

Як керуючий сигнал блоку **If** використовується генератор сигналу прямокутної форми **Pulse Generator** (розділ Sources), що генерує Input Signal (див. рис. 5.4.). Коли Input Signal > 0, працює **Action Block** відповідний then, а саме ABS Output Reset, у протилежному випадку працює **Action Block** відповідний else, а саме **Saturation between -0.75 and 0.75 Output held**. На обидва ці блоки подається синусоїдальний сигнал, що обробляється відповідно до алгоритмів блоків **ABS** або **Saturation** залежно від стану **Input Signal**.

При **Input Signal** > 0 синусоїдальний сигнал блоком **ABS** перетворюється в сигнал **Output Reset** (негативні значення перемножуються на -1), потім блок **Sum** додає до значень цього сигналу 2, а до синусоїдального 0 (див. рис. 4.4. сигнал **Output Reset**).

При **Input Signal** ≤ 0 блок **Saturation** пропускає тільки значення сигналу, що знаходяться в інтервалі від -0.75 до 0.75. Потім блок **Sum** додає до значень цього сигналу 2 (див. рис. 4.4. сигнал Output held).

Умова виконання гілок then й else блоку If задаються в параметрах блоку.

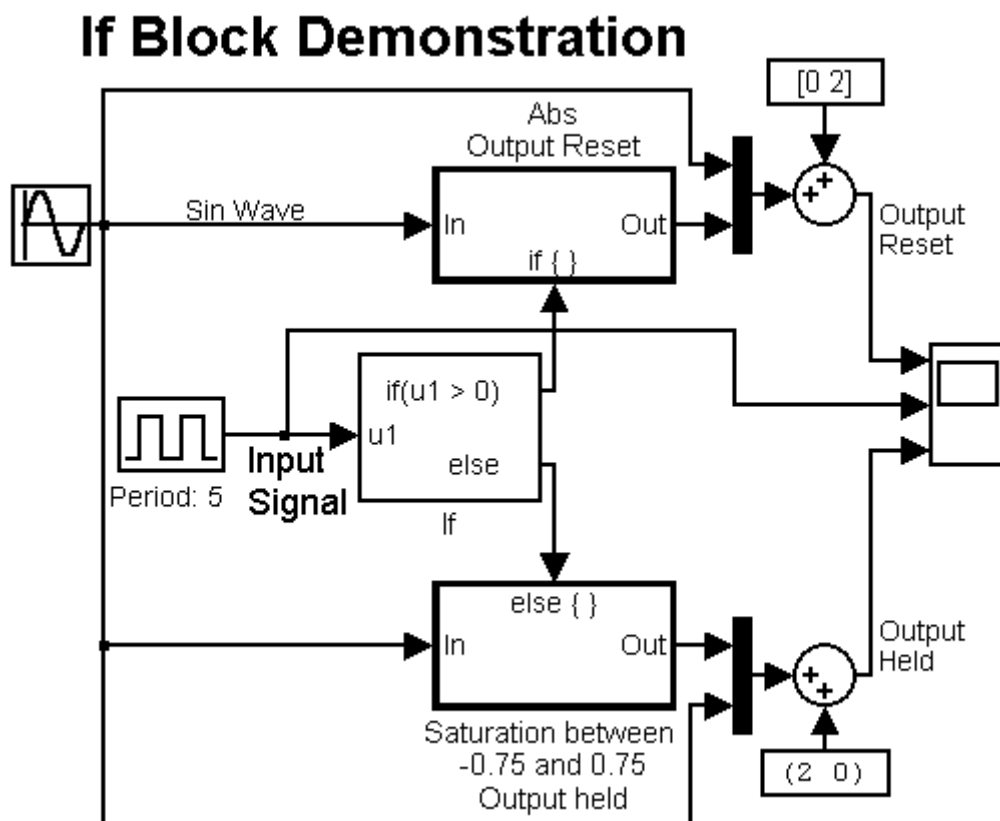


Рис. 5.1. Приклад використання блоку If.

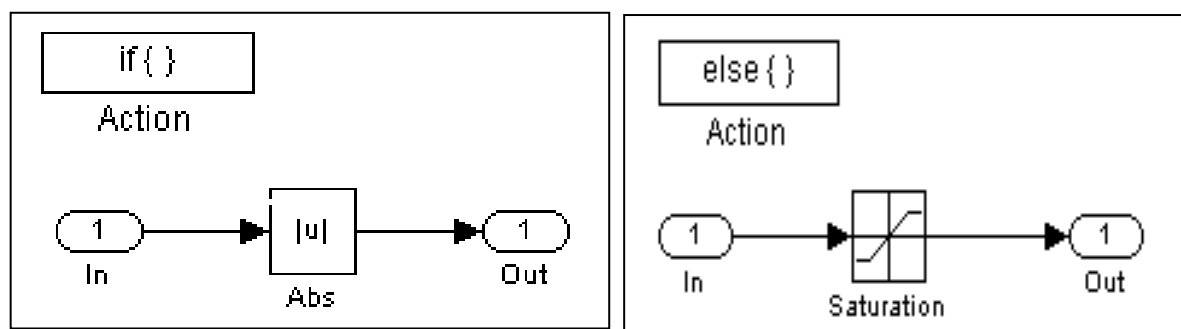


Рис. 5.2. Содержимое блока

Рис. 5.3. Содержимое блока and

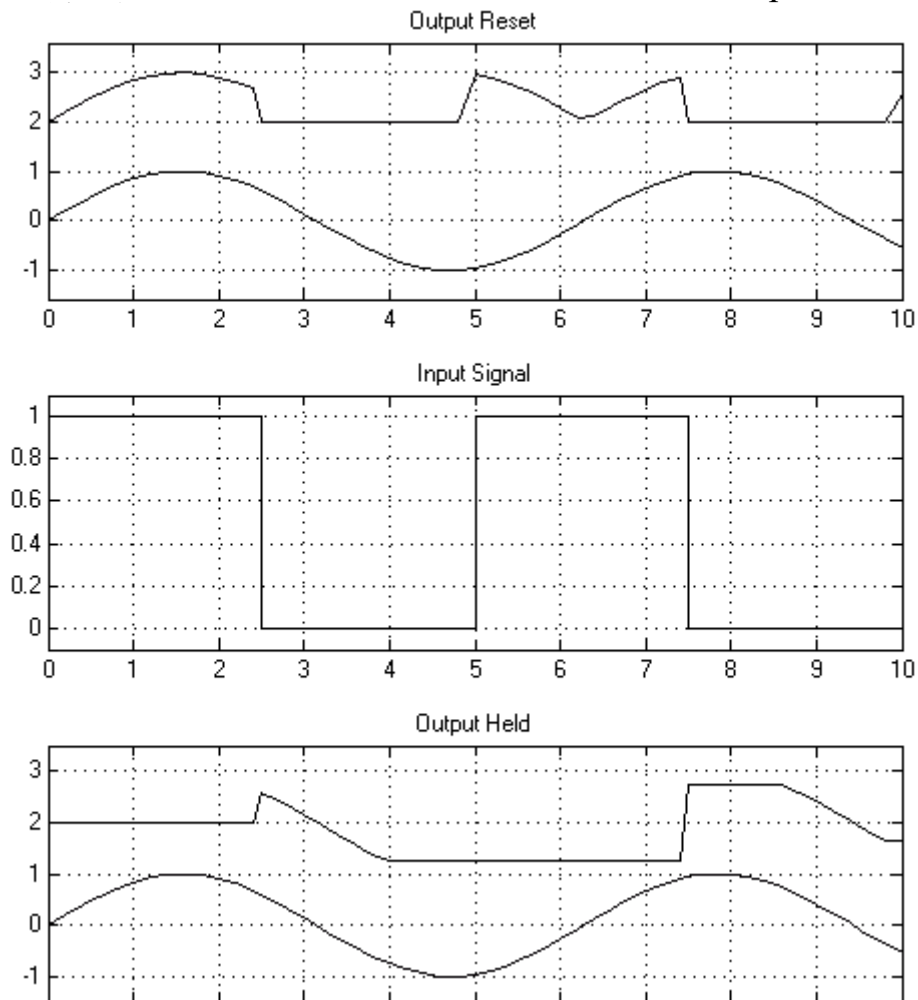


Рис. 5.4. Вікно Score блок-діаграми приклада використання блоку If.

Контрольні запитання

1. Поясніть призначення блоку Relational Operator, які параметри блоку?
2. Поясніть призначення блоку Combinatorial Logic, які параметри блоку?
3. Поясніть призначення блоку IF, які параметри блоку?

ЛАБОРАТОРНА РОБОТА № 6

МОДЕЛЮВАННЯ ТА АНАЛІЗ ВИПАДКОВИХ ЗБУРЕНЬ, ЩО ВПЛИВАЮТЬ НА РОБОТУ ПРИЛАДІВ

Мета роботи: Ознайомитись з роботою блоків Uniform Random Number, Hit Crossing, Logical Operator, Discrete Time Integrator.

Завдання до лабораторної роботи

1. Скласти блок-діаграму, що забезпечує реєстрацію кількості фактів влучення випадкових чисел у заданий інтервал часу (згідно варіанта) за певний час спостереження, вивести значення моментів часу влучення і самого значення випадкового числа в робочу область MatLab. У блок Scope в одному вікні вивести значення випадкової величини, а в другому - сигнал, що ідентифікує подію влучення випадкової величини в заданий інтервал.
2. В отриманій блок-діаграмі джерело випадкових чисел Uniform Random Number змінити на MATLAB Fcn.
3. Перетворити блок-діаграму так, щоб обчислювався час до першого влучення випадкових чисел у заданий інтервал, і після цього відбувалася зупинка роботи моделі

Варіанти завдань до лабораторної роботи №6.

№	Інтервал, у якому рівномірно розподілені випадкові числа	Інтервал влучення, у який випадкових чисел необхідно реєструвати	Час спостере- ження
1	1...10	2...6	9
2	0...12	3...10	10
3	0...15	5...11	9
4	2...10	3...5	10
5	0...18	2...7	9
6	3...10	4...5	10
7	5...18	6...15	9

Варіанти завдань до лабораторної роботи №6.

№	Інтервал, у якому рівномірно розподілені випадкові числа	Інтервал влучення, у який випадкових чисел необхідно реєструвати	Час спостере- ження
8	0...15	3...10	10
9	1...17	3...5	9
10	0...13	1...2	10

Рекомендації для виконання пункту 1 завдання.

1. За допомогою блоку Uniform Random Number (розділ Sources) сформувати вхідний сигнал, що представляє собою випадкові числа рівномірно розподілені в інтервалі згідно завдання.
2. За допомогою двох блоків Hit Crossing (розділ Signals&Systems), а також блоку Logical Operator (розділ Math) організувати з'ясування факту влучення випадкової величини в заданий інтервал.
3. Підрахунок кількості фактів влучення організувати за допомогою блоку Discrete Time Integrator (розділ Discrete), який виконує підсумовування інтервалів часу надходження вхідного сигналу 1 від блоку Logical Operator. На виході блоку Discrete Time Integrator з'являється одиничний сигнал, коли на вхід поступає одиничний сигнал причому на виході Discrete Time Integrator буде підтримуватись значення 1 на протязі кроку моделювання, який заданий параметром Sample time блоку. Тобто сумарне значення на виході в Discrete Time Integrator рівне добуткові кількості влучень випадкової величини в заданий інтервал на крок моделювання, що заданий параметром Sample time блоку. Тому для підрахунок кількості фактів доцільно встановити крок моделювання (параметр Sample time блоку Discrete Time Integrator) рівним 1.
4. Кількість фактів влучення випадкової величини в заданий інтервал отриманий на виході блоку Discrete Time Integrator вивести на Display.

5. У блоці Score в одному вікні вивести значення випадкової величини, а в другому - сигнал, що ідентифікує подію влучення випадкової величини в заданий інтервал.
6. За допомогою блоку Mux об'єднайте у вектор сигнали:
- значення поточного часу одержане від блоку Clock;
 - значення випадкової величини на попередньому кроці моделювання, одержане за допомогою блоку Memory;
 - поточне значення випадкової величини помножене на вихідний сигнал блоку Logical Operator за допомогою блоку Product;
 - вихідний сигнал блоку Discrete Time Integrator.
7. Отриманий вектор подайте на вхід блоку To Workspace та встановіть параметри блоку згідно рис. 6.1. Зверніть увагу, що ім'я змінної робочої області Workspace, у якій будуть зберігатися значення сигналів об'єднаних у вектор Ivan. Задайте інше ім'я змінної.

- To Workspace -

Write input to specified array or structure in MATLAB's main workspace. Data is not available until the simulation is stopped or paused.

Параметры

Variable name:

Ivan

Limit data points to last:

inf

Decimation:

1

Sample time (-1 for inherited):

-1

Save format: Array

Рис. 6.1. Параметри настроювання блоку To Workspace.

8. Встановіть час спостереження й періодичність реєстрації результатів. Прослідкуйте, щоб час моделювання був більшим ніж час спостереження на крок моделювання блоку Discrete Time Integrator (параметр Sample time) на той випадок щоб блоком Discrete Time Integrator був врахований факт влучення, який стався на останній момент часу спостереження. Параметр Sample time всіх інших блоків моделі доцільно встановити рівним (-1).
9. Після здійснення моделювання в робочій області Workspace введіть ім'я змінної, при цьому Ви повинні отримати данні аналогічні приведеним на

```
>> Ivan

Ivan =

      0      0      0      0
0.2569 6.9279 1.0000 0
6.9279      0 2.0000 1.0000
7.5344      0 3.0000 1.0000
9.5356 3.8732 4.0000 1.0000
3.8732      0 5.0000 2.0000
0.5616 3.7534 6.0000 2.0000
3.7534      0 7.0000 3.0000
6.5751      0 8.0000 3.0000
7.0177 0.9074 9.0000 3.0000
0.9074 9.4799 10.0000 4.0000
9.4799 5.3485 11.0000 5.0000

>>
```

рис. 6.2.:

Рис. 6.2. Одержання значення змінної Ivan робочої області Workspace

10. Порівняйте значення отримане у блоці Display із значеннями змінної робочої області Workspace. Поясніть отримані результати.

Рекомендації для виконання пункту 2 завдання.

В отриманій блок-діаграмі джерело випадкових чисел Uniform Random Number змінити на MATLAB Fcn (функція з розділу Functions&Tables).

Рекомендації для виконання пункту 3 завдання.

Для зупинки процесу моделювання після першого влучення випадкових чисел у заданий інтервал потрібно у створену раніше блок-діаграму додати

блок Stop Simulation на вхід якого подати вихідний сигнал блоку Logical Operator. Для обчислення часу до першого влучення випадкових чисел у заданий інтервал можна використати блок Clock, вихідний сигнал якого подати на Display, де отримаємо значення часу.

Звіт по виконанню роботи повинні містити:

- Блок-діаграму, що забезпечує реєстрацію кількості фактів влучення випадкових чисел у заданий інтервал часу.
- Значення моментів часу влучення і відповідних значень випадкового числа з робочої області MatLab.
- Значення параметрів всіх блоків і параметри моделювання Simulink, що забезпечують час спостереження й періодичність реєстрації результатів згідно варіанта.
- Вигляд вікна Scope із двома графіками в різних координатних осях, а саме: значення випадкової величини й сигнал, що ідентифікує факт влучення випадкової величини в заданий інтервал.
- Блок-діаграму з джерелом випадкових чисел від блоку MATLAB Fcn (розділ Functions&Tables) і параметри блоку.
- Пояснення, яким чином задавалися:
 - інтервал рівномірного розподілу випадкових чисел;
 - інтервал, влучення в який, випадкових чисел необхідно реєструвати;
 - сумарний час спостереження;
 - період реєстрації результатів.
- Перетворену блок-діаграму для обчислення часу до першого влучення випадкових чисел у заданий інтервал і зупинки роботи моделі після

першого влучення випадкових чисел у заданий інтервал (пункт завдання 2).

Контрольні запитання

1. Поясніть призначення блоку Uniform Random Number, які параметри цього блоку?
2. Поясніть призначення блоку Hit Crossing, які параметри цього блоку?
3. Поясніть призначення блоку Logical Operator, які параметри цього блоку?
4. Поясніть призначення блоку Discrete Time Integrator, які параметри цього блоку?
5. Пояснення, яким чином задавався інтервал рівномірного розподілу випадкових чисел.
6. Пояснення, яким чином інтервал, влучення в який, випадкових чисел необхідно реєструвати.
7. Пояснення, яким чином сумарний час спостереження.
8. Пояснення, яким чином період реєстрації результатів.

ЛАБОРАТОРНА РОБОТА №7

ДИНАМІЧНИЙ АНАЛІЗ СИСТЕМ АВТОМАТИЧНОГО КЕРУВАННЯ

Мета роботи: Навчитися створювати блок-модель для побудови статичної та перехідної характеристик шляхом моделювання за допомогою Simulink. Засвоїти методику побудови перехідної, імпульсної перехідної та частотних характеристик за допомогою вікна LTI Viewer Control System Toolbox та за допомогою функцій Control System Toolbox, навчитися отримувати аналітичні вирази та будувати перехідну й імпульсну характеристики за допомогою Symbolic Math.

Завдання до лабораторної роботи

Для типового коригуючого пристрою (табл.1, група 1 коригуючих пристроїв) згідно варіанта, виконати наступне:

1. Створити блок-модель для побудови статичної характеристики шляхом моделювання за допомогою Simulink.
2. Створити блок-модель для побудови перехідної характеристики шляхом моделювання за допомогою Simulink.
3. За допомогою вікна LTI Viewer Control System Toolbox побудувати:
 - перехідну характеристику,
 - імпульсну перехідну (вагову) характеристику,
 - амплітудну частотну характеристику,
 - фазову частотну характеристику,
 - амплітудно-фазову характеристику,
 - логарифмічну амплітудну частотну характеристику,
 - логарифмічну фазову частотну характеристику,
 - нулі й полюси передатної функції,

забезпечивши позначення на графіках сітки й характерних точок.

4. Побудувати динамічні і частотні характеристики за допомогою функцій Control System Toolbox.
5. Одержати аналітичні вирази й побудувати перехідну й імпульсну характеристики за допомогою Symbolic Math.
6. Створити блок-діаграми для одержання перехідної й імпульсної характеристик, використовуючи як вхідний сигнал генератор прямокутних імпульсів Pulse Generator, установивши його параметри відповідним чином.
7. Створити блок-діаграму, за допомогою якої для 2-х значень частоти одержати вхідний і вихідний сигнали й показати відповідні точки на

графіках частотних характеристик (тобто продемонструвати технологію побудови частотних характеристик).

8. Для типового коригуючого пристрою (табл.1, група 2 коригуючих пристроїв) одержати вхідний і вихідний сигнали, а також логарифмічну амплітудну частотну й логарифмічну фазову частотну характеристики. Пояснити відмінність у роботі двох коригуючих пристроїв і відображення цих відмінностей у їхніх характеристиках.

9. Для типової ланки (табл. 2) згідно варіанту виконати пункти 1-7 завдання.

10. Для системи автоматичного регулювання (табл.3) згідно варіанту виконати пункти 1-7 завдання.

Звіт по виконанню роботи повинні містити:

Для типового коригуючого пристрою, типової ланки й системи автоматичного регулювання привести:

1. Блок-діаграми, динамічні й частотні характеристики з позначенням характерних точок і поясненням їхнього сенсу.
2. Команди, що задаються у командному вікні MatLab, для побудови динамічних і частотних характеристик за допомогою функцій Control System Toolbox.
3. Команди, що задають у командному вікні MatLab, для побудови динамічних характеристик за допомогою Symbolic Math.
4. Аналітичні вирази перехідної й імпульсної характеристик.
5. Перехідна й імпульсна характеристики, отримані з використанням блоку Pulse Generator. Привести параметри блоку Pulse Generator і параметри Simulink, встановлені для побудови характеристик. Порівняти отримані графіки з отриманими раніше.

6. Блок-діаграма й графіки залежності вихідного сигналу від вхідного для двох значень частоти. Пояснити технологію побудови частотних характеристик.

11. Вхідний, вихідний сигнали, логарифмічну амплітудної частотної й логарифмічної фазову частотну характеристики типового коригуючого пристрою (табл.1, група 2 коригуючих пристроїв). Пояснити відмінність у роботі двох коригуючих пристроїв і відображення цих відмінностей у їхніх характеристиках.

12.Визначення передатних функцій двох коригувальних пристроїв.

Динамічні і частотні характеристики системи.

Безперервна система автоматичного керування (САК):

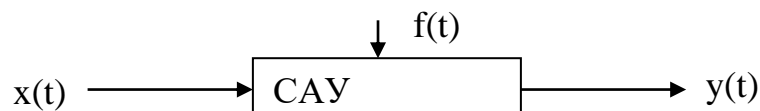


Рис. 7.1. Вхідний $x(t)$, вихідний $y(t)$ і збурюючий $f(t)$ сигнали САК.

Через інерційність САК, процеси в ній описуються диференціальними рівняннями n -го порядку з постійними коефіцієнтами. Тому в узагальненому виді зв'язок між вхідним і вихідним сигналом описується рівнянням:

$$\begin{aligned}
 & a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \frac{dy}{dt} + a_0 y = \\
 & = b_m \frac{d^m x}{dt^m} + b_{m-1} \frac{d^{m-1} x}{dt^{m-1}} + \dots + b_1 \frac{dx}{dt} + b_0 x.
 \end{aligned}$$

Для зручного запису диференціальні рівняння записують в операторній формі, уводячи оператор диференціювання.

$$\frac{d}{dt} = p; \quad \frac{dx(t)}{dt} = px(t); \quad \frac{1}{p} = \int_0^t ; \quad \frac{x(t)}{p} = \int_0^t x(t)dt.$$

Застосувавши цей оператор до диференціального рівняння САК, одержимо диференціальне рівняння в операторній формі:

$$(a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0) y(t) = (b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0) x(t).$$

Позначимо:

$A(p) = (a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0)$ – оператор при вихідній величині;

$B(p) = (b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0)$ – оператор при вхідній величині.

Одержимо:

$$A(p) \cdot y(t) = B(p) \cdot x(t).$$

Якщо враховувати залежність $y(x, f)$, то одержимо:

$$A(p) \cdot y(t) = B(p) \cdot x(t) + C(p) \cdot f(t).$$

$$y(t) = \frac{B(p)}{A(p)} x(t), \text{ де } \frac{B(p)}{A(p)} = W(p) \text{ – операторна передатна функція.}$$

$$y(t) = W(p) \cdot x(t); \quad x(t) \longrightarrow \boxed{W(p)} \longrightarrow y(t)$$

Рис. 7.2. Система, представлена операторною передатною функцією.

Операційна передатна функція заснована на застосуванні перетворення Лапласа. Суть операційного числення полягає в тому, що якщо маємо $x(t)$, то їй протиставляють $X(s)$, де $s = \alpha + j\omega$ – комплексна змінна. $x(t)$ – оригінал, а $X(s)$ – зображення.

$$\text{Перетворення Лапласа: } X(s) = L[x(t)] = \int_{-\infty}^{+\infty} x(t) e^{-st} dt$$

Зворотне перетворення Лапласа: $x(t)=L^{-1}[X(s)]$.

Тепер функції часу можна замінити їхніми зображеннями. Зв'язує ці зображення передатна функція $W(s)$:

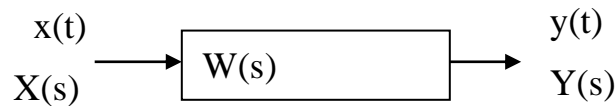


Рис. 7.3. Система, представлена передатною функцією.

Передатна функція $W(s)$ це відношення зображення вихідної величини до зображення вхідної величини при нульових початкових умовах:

$$W(s) = \frac{Y(s)}{X(s)}; \quad Y(s) = W(s) \cdot X(s).$$

Приклад. Рівняння електродвигуна:

$$U(t) \rightarrow \boxed{W(s)} \rightarrow \alpha(t) \quad T \frac{d^2 \alpha(t)}{dt^2} + \frac{d\alpha(t)}{dt} = k U(t), \quad T, k - \text{const.}$$

Запишемо вихідне диференціальне рівняння в операційній формі (по Лапласу) при нульових початкових умовах:

$$Ts^2 \alpha(s) + s \alpha(s) = k U(s).$$

Звідки вихідна величина: $\alpha(s) = \frac{kU(s)}{s(Ts + 1)}.$

Передатна функція: $W(s) = \frac{\alpha(s)}{U(s)} = \frac{kU(s)}{s(Ts + 1)U(s)} = \frac{k}{s(Ts + 1)}.$

Передатна функція $W(s) = \frac{B(s)}{A(s)}$ має нулі і полюси. Нулі (такі значення аргументу s , при яких $W(s)=0$). Полюси (значення s , при яких $W(s)=0$). Для знаходження нулів потрібно розв'язати рівняння $B(s)=0$, а

поліосів - $A(s)=0$. Нулі й поліоси графічно зображуються на комплексній площині:

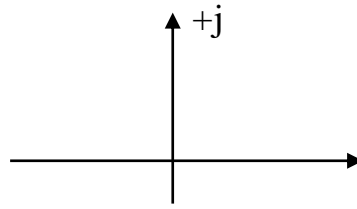


Рис. 7.4. Комплексна площина для зображення нулів і поліосів передатної функції.

Для оцінки динамічних властивостей системи використовуються **динамічні характеристики** (функції):

- **перехідна функція** $h(t)$ – реакція системи, а саме вихідний сигнал y , на вхідний вплив типу одиничного стрибка $1(t)$ при нульових початкових умовах (вважають, що система перебувала до впливу в стані спокою).

Для перехідної характеристики, враховуючи, що $Y(s)=W(s) \cdot X(s)$ справедливо наступне:

- вхідний вплив: $x(t)=1(t)$; його зображення: $X(s)=\frac{1}{s}$;
- вихідний сигнал: $y(t)=h(t)$; його зображення: $Y(s)=H(s)$;
- зображення перехідної характеристики: $H(s)=W(s) \cdot X(s)=\frac{W(s)}{s}$;
- перехідна характеристика: $h(t)=L^{-1}(H(s))=L^{-1}\left(\frac{W(s)}{s}\right)$.

- **імпульсна характеристика (функція ваги)** $g(t)$ – реакція системи на зовнішній вплив дельта-функції $\delta(t)$.

Для імпульсної характеристики, враховуючи, що $Y(s)=W(s) \cdot X(s)$ справедливо наступне:

- вхідний вплив: $x(t)=\delta(t)$; його зображення: $X(s)=1$;
- вихідний сигнал: $y(t)=g(t)$; його зображення: $Y(s)=G(s)$;

- зображення імпульсної характеристики: $G(s)=W(s) \cdot X(s)=W(s)$;
- імпульсна характеристика: $g(t)=L^{-1}(G(s))=L^{-1}(W(s))$.

Типові впливи зв'язані між собою співвідношенням: $\delta(t)=\frac{dl(t)}{dt}$.

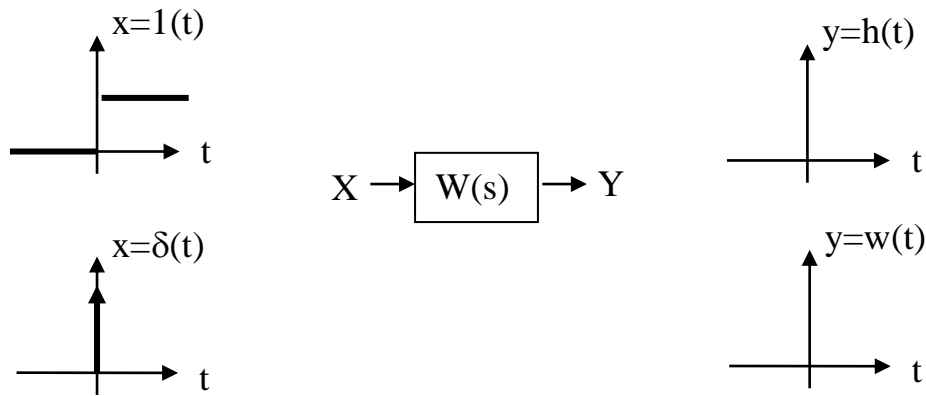


Рис. 7.5. Типові входні впливи, перехідна й імпульсна характеристики системи.

Одними йз основних характеристик системи є також частотні характеристики. При входньому синусоїдальному впливі виду $x=A_{\text{вх}}\sin(\omega t+\varphi_0)$, де ω – частота; $A_{\text{вх}}$ – амплітуда; φ_0 – початкова фаза, необхідно довідатися сигнал на виході. Для цього застосовуються частотні характеристики.

$$y(t) = y_{\text{вільне}}(t) + y_{\text{вимушене}}(t).$$

У стійкій системі $\lim_{t \rightarrow \infty} y_{\text{вільне}}(t)=0$, тому: $y_{\text{встановлене}}=y_{\text{вимушене}}(t)$, а отже під

вихідним сигналом $y(t)$ розуміють $y_{\text{вимушене}}(t)$:

$$y(t) = y_{\text{вимушене}}(t) = A_{\text{вих}}\sin(\omega t+\varphi).$$

Тобто вихідний сигнал має ту ж частоту, що й вхідний, але він відрізняється по амплітуді від вхідного й має зсув фаз.

Частотні характеристики застосовуються для того, щоб знаючи частоту ω , і амплітуду $A_{\text{вх}}$, вхідного синусоїдального сигналу $x=A_{\text{вх}}\sin(\omega t)$ знайти амплітуду вихідного сигналу $A_{\text{вих}}$, і зсув фаз φ вхідного й вихідного синусоїдального сигналу $y(t)=A_{\text{вих}}\sin(\omega t+\varphi)$:

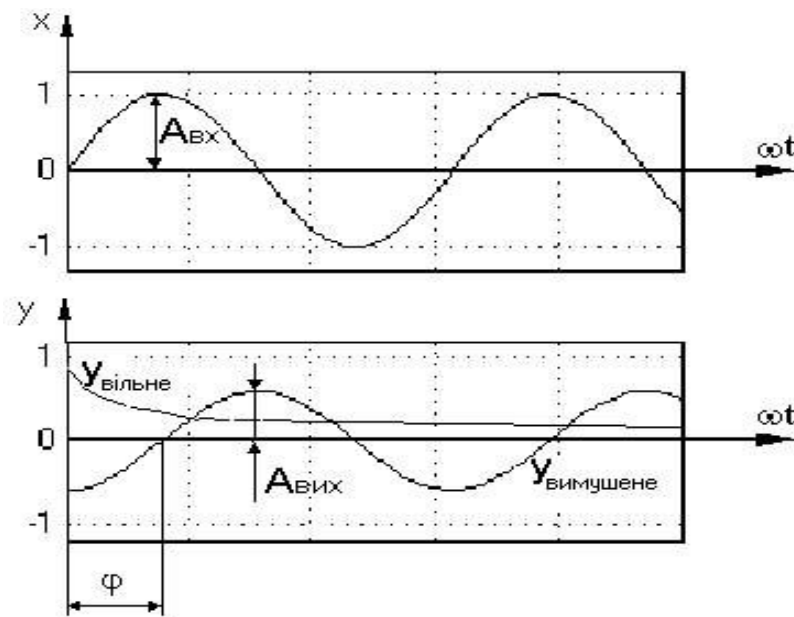


Рис. 7.6. Вхідний синусоїдальний сигнал $x=A_{\text{вх}}\sin(\omega t)$ і вихідний синусоїдальний сигнал $y(t)=A_{\text{вих}}\sin(\omega t+\varphi)$ системи.

Розглядається п'ять частотних характеристик:

1) **Амплітудно-фазова** (комплексна передатна функція):

$$W(j\omega) = \frac{A_{\text{вих}} e^{j(\omega t + \varphi)}}{A_{\text{вх}} e^{j\omega t}} = \frac{A_{\text{вих}}}{A_{\text{вх}}} e^{j\varphi}.$$

2) **Амплітудна частотна характеристика** – залежність відношення амплітуди виходу до амплітуди входу від частоти:

$$W(\omega) = \frac{A_{\text{вих}}}{A_{\text{вх}}}.$$

3) **Фазова частотна характеристика** – залежність зсуву фаз між вихідним і вхідним сигналами від частоти:

$$\varphi = \varphi(\omega).$$

Логарифмічні частотні характеристики - це частотні характеристики побудовані в логарифмічних масштабах, вони застосовуються через простоту їхньої побудови.

4) **Логарифмічна амплітудна частотна характеристика:**

$$L(\omega) = 20 \lg W(\omega)$$

$W(\omega) = \frac{A_{\text{ВЫХ}}}{A_{\text{ВХ}}}$	100	10	5	2	1,13	1	0,01
$L(\omega)$, децибел	+40	+20	+14	+6	+1	0	-40

5) Логарифмічна **фазова частотна характеристика** розраховується по такій же формулі, як звичайна.

Побудова перехідної характеристики шляхом моделювання за допомогою Simulink.

Для прикладу розглянемо аперіодичну ланку першого порядку, передатна функція якого наступна:

$$W(s) = \frac{4}{0,5s + 1}.$$

Для одержання перехідної характеристики необхідно побудувати модель системи, до входу підключити блок одиничного стрибка Step, а до виходу - блок осцилографа Scope.

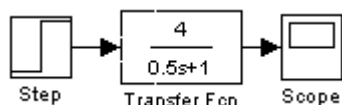


Рис. 7.7. Блок-модель для одержання перехідної характеристики.

При аналізі параметрів перехідного процесу необхідно враховувати, що за замовчуванням у блоці Step час початку стрибка – 1с, а не 0с.

Імпульсну характеристику одержати за допомогою Simulink складніше, тому що блок, що формує δ -функцію, відсутній, а його моделювання шляхом диференціювання одиничного стрибка дає велику похибку.

Побудова динамічних і частотних характеристик за допомогою вікна LTI Viewer Control System Toolbox

Simulink має динамічний зв'язок з Control System Toolbox - пакетом, що дозволяє досліджувати динамічні характеристики систем.

У вікні, що відкривається, меню Tools вибирається опція Linear Analysis, при цьому з'являється додаткова бібліотека вхідних і вихідних портів Model Inputs and Outputs і порожнє вікно блоку LTI Viewer. Вхідні й вихідні порти, як звичайні бібліотечні блоки, перетаскуються у вікно моделі й приєднуються до досліджуваної системи:

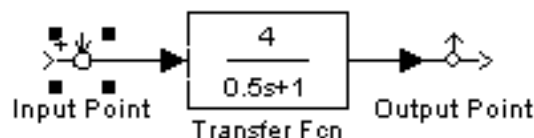


Рис. 7.8. Блок-модель для одержання динамічних і частотних характеристик за допомогою вікна LTI Viewer Control System Toolbox.

Далі для проведення моделювання в меню, що випадає, вікна Simulink вибирається опція Get Linearized Model.

У вікні LTI Viewer вибирається опція Plot Configuration з падаючого меню Edit й у вікні, що відкрилося, задаються необхідні для аналізу характеристики (графіки) системи:

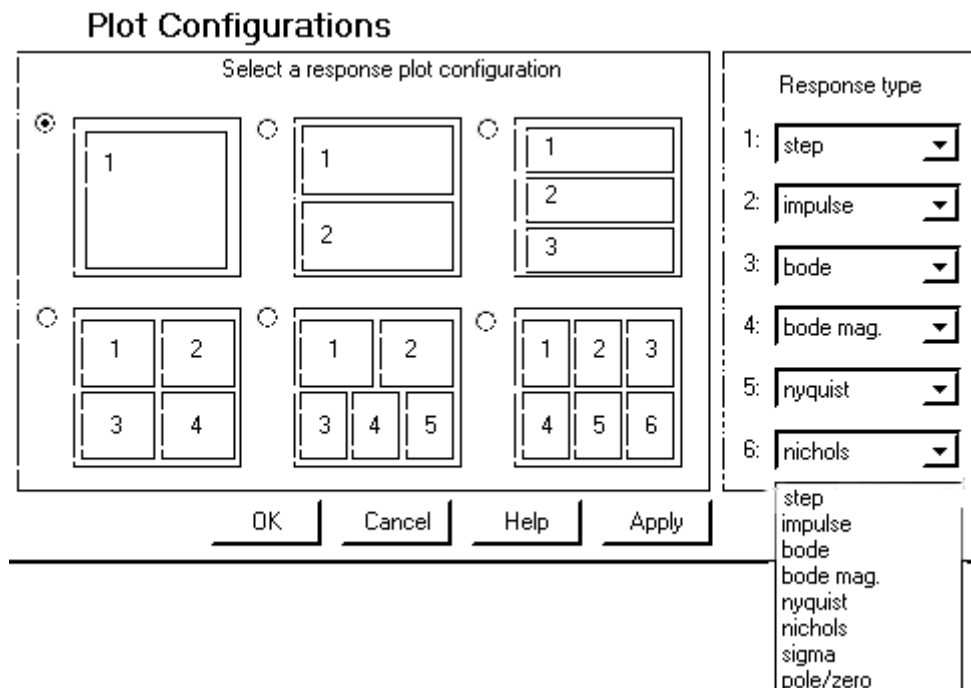


Рис. 7.9. Вікно настроювання відображення результатів моделювання.

Опції відображення графіків задаються в меню Edit->Viewer Preferences:

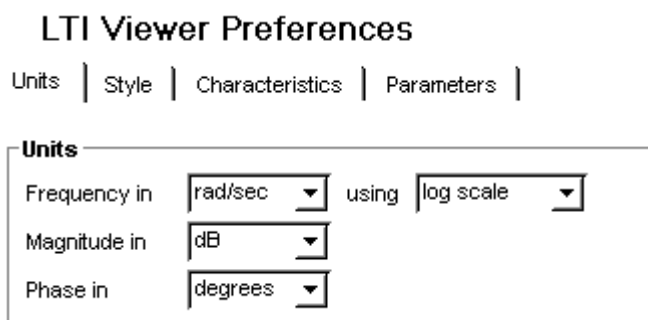


Рис. 7.10. Вікно завдання опцій виводу графіків.

У вкладці **Units** вікна LTI Viewer Preferences:

- Magnitude in... - масштаб абсолютної величини (dB-децибели, absolute - абсолютний)
- Phase in... - масштаб фази (radians - радіани, degrees - градуси)
- Frequency in... - масштаб частоти (Rad/sec - радіани в секунду, Hz - герци).
- Using - log scale - логарифмічну, lineal scale - лінійну шкалу.

У вкладці **Characteristics** вікна LTI Viewer Preferences:

- Show setting time within a% - обчислювати час регулювання для +a% щодо сталого значення.
- Show rise time from p% to y% - обчислювати час наростання від p% до y% щодо сталого значення.

У вкладці **Parameters** вікна LTI Viewer Preferences задаються параметри виводу динамічних характеристик:

Time Vector - вектор часу:

- Generate automatically - генерувати автоматично;
- Define vector - задати початкове значення, приріст, кінцеве значення.

Frequency Vector - вектор частот:

- Generate automatically генерувати автоматично;
- Define vector - задати в логарифмічному масштабі показник степені 10 нижньої частоти, показник степені 10 верхньої частоти, N - кількість точок.

У вкладці **Style** вікна LTI Viewer Preferences задаються параметри виду ліній, якими виводяться графіки, порядок застосування кольорів, маркування й стилю лінії для поділу систем, входів, виходів і зв'язків.

За допомогою контекстного меню, яке відкривається за допомогою правої кнопки миші, для кожного із графіків можна задати його оформлення: кольори, масштабування, наявність сітки, також наявність найбільш характерних точок графіків, наприклад, таких як час регулювання, перерегулювання, запаси стійкості й т.п.

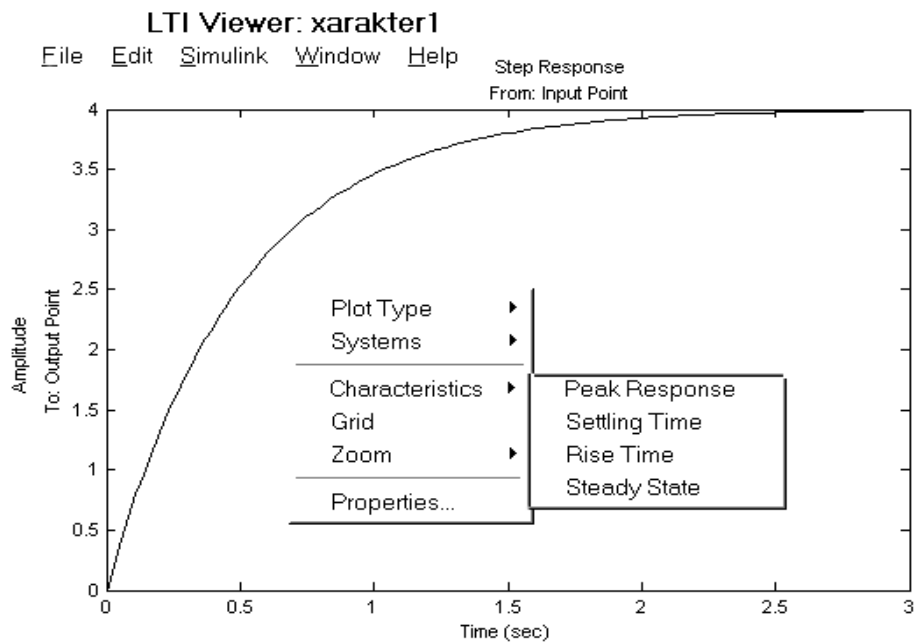


Рис. 7.11. Контекстне меню графіка.

Команди контекстного меню графіка:

Plot type - виведена характеристика

Systems - системи, для яких будуються характеристики

Characteristics - відображення характерних точок графіка;

Zoom - масштабування;

Grid - виводити сітку на графік.

Імена моделей, доступних для аналізу, виводяться в меню Systems.

Приклади графіків отриманих за допомогою вікна LTI Viewer Control System Toolbox:

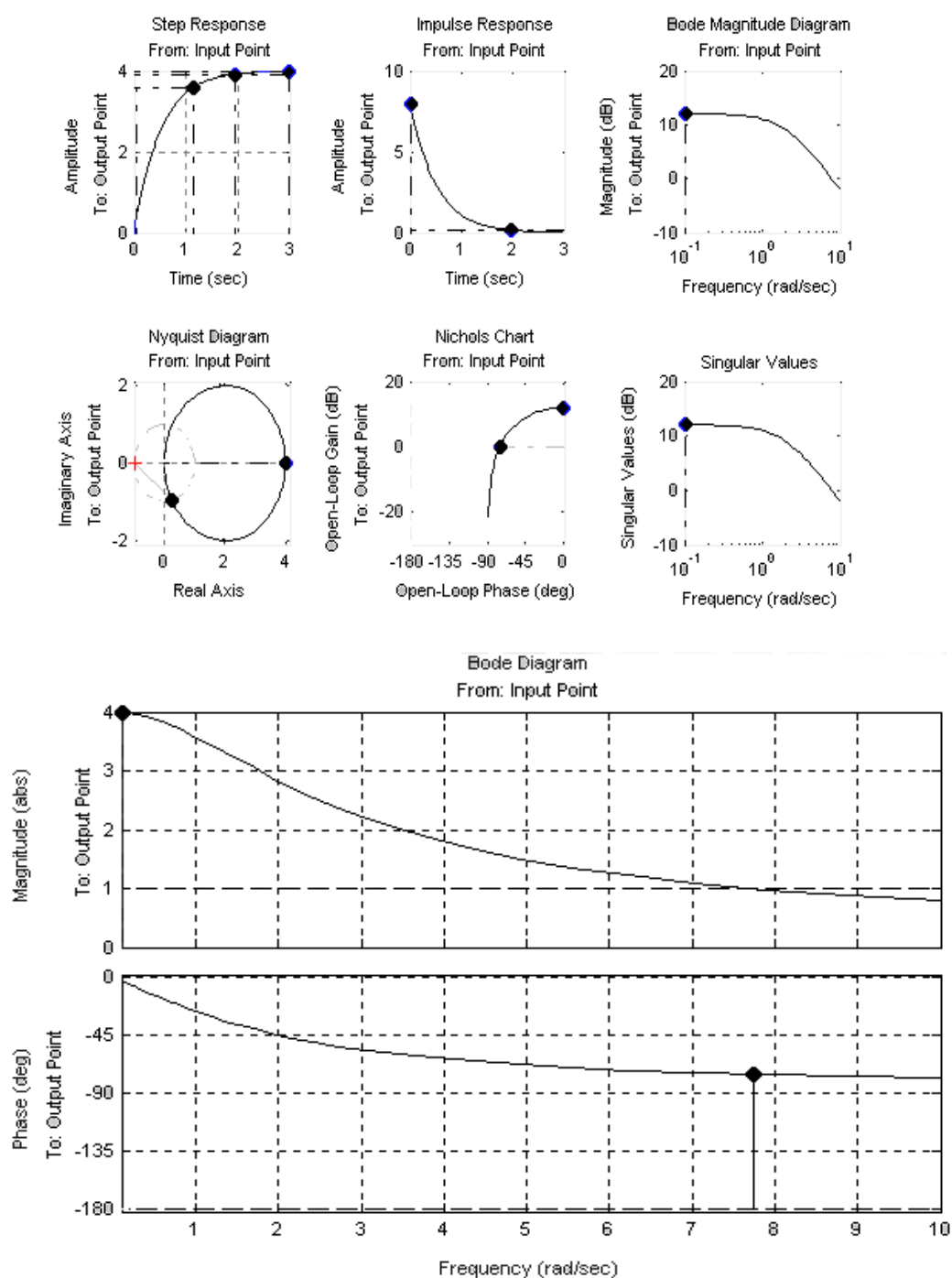


Рис. 7.12. Приклади графіків отриманих за допомогою вікна LTI Viewer Control System Toolbox:

Побудова динамічних і частотних характеристик за допомогою функцій Control System Toolbox

Спочатку необхідно задати опис системи. Це можна зробити одним із чотирьох способів:

1. У вигляді передатної функції за допомогою функції `tf`:

```
>> sys=tf([4], [0.5 1])  
Transfer function:  
      4  
-----  
0.5 s + 1
```

Параметрами функції `tf` є вектора коефіцієнтів чисельника й знаменника.

2. У вигляді полюсів, нулів і коефіцієнта передачі передатної функції за допомогою функції `zpk`:

```
>> sys=zpk([], [-2], 8)  
Zero/pole/gain:  
      8  
-----  
(s+2)
```

Параметрами функції `zpk` є вектори нулів, полюсів і коефіцієнт передачі.

3. У просторі стану за допомогою функції `ss`:

```
>> sys=ss([-0.5], [4], [2], [0])
```

```
a =  
      x1  
x1 -0.5  
b =
```

```

    u1
x1  4
c =
    x1
y1  2
d =
    u1
y1  0
Continuous-time model.

```

Параметрами функції ss є матриці стану системи A, B, C, D.

4. Створити в Simulink блок-модель наступного виду:

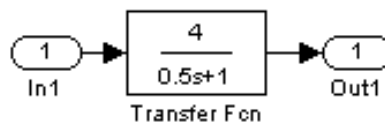


Рис. 7.13. Блок-модель для одержання інформацію про моделі.

Потім отримати інформацію про цю модель, вказавши її назву, як ім'я файлу, у даному прикладі 'untitled':

```

>> [A, B, C, D]=linmod('untitled')
A =
    -2.0000
B =
     1.0000
C =
     8.0000
D =
     0

```


Функції `linmod` повертає опис моделі в просторі стану за допомогою матриць стану A , B , C , D . Параметром функції `linmod` є ім'я моделі. Далі можна перетворити матриці стану в модель Control System:

```
>> sys=ss(A, B, C, D)
```

```
a =
```

```
    x1
```

```
    x1 -2
```

```
b =
```

```
    u1
```

```
    x1  1
```

```
c =
```

```
    x1
```

```
    y1  8
```

```
d =
```

```
    u1
```

```
    y1  0
```

```
Continuous-time model.
```

Отримана модель може бути використана для побудови динамічних і частотних характеристик системи за допомогою відповідних функцій:

- » `step(sys)` – перехідна характеристика;
- » `impulse(sys)` – імпульсна характеристика;
- » `bode(sys)` - логарифмічні амплітудні й фазова частотні характеристики;
- » `nyquist(sys)` – амплітудно-фазова частотна характеристика.

Параметром функцій являється ім'я опису системи (передатної функції). При цьому діапазон частот для побудови графіків вибирається автоматично. Можна задати діапазон частот, наприклад команда `>>`

bode(sys, {0.01 1000}) побудує характеристику в діапазоні частот 0.01 - 1000Гц.

Задавши опис системи одним зі способів, можна також за допомогою відповідних функцій одержати опис системи у вигляді:

- передатної функції - функція `tf(sys)`;
- нулів/полісів/коефіцієнта передачі - функція `zpk(sys)`;
- у просторі стану (функція `ss(sys)`,

де `sys` - назва змінної, яка ідентифікує досліджувану систему. Наприклад,

```
>> tf(sys)
```

Transfer function:

8

s + 0.5

Побудова динамічних характеристик й одержання їхнього аналітичного вираження за допомогою Symbolic Math

У командному вікні MatLab ініціалізуємо символ `s` і значення передатної функції привласнимо змінній:

```
>> syms s
```

```
>> w=4/(0.5*s+1)
```

w =

```
4/(1/2*s+1)
```

Обчислення зворотного перетворення Лапласа від зображення динамічної характеристики для перехідної характеристики:

```
>> ilaplace(w/s)
```

```
ans =  
4-4*exp(-2*t)
```

Обчислення зворотного перетворення Лапласа від зображення тимчасової характеристики для імпульсної характеристики:

```
>> ilaplace(w)  
ans =  
8*exp(-2*t)
```

Результатом виконання функції `ilaplace` є аналітичний вираз відповідної динамічної функції. Знаючи аналітичний вираз функції, можна побудувати її графік за допомогою функції `plot` наступним чином:

Задати інтервал часу для побудови графіків функцій:

```
>> t=(0:0.1:10)           0-10 - часовий інтервал;   0,1 - крок зміни.
```

Як перший параметр функції `plot` необхідно вказати вектор, що задає часовий інтервал для побудови графіка, у якості другого – отриманий раніше аналітичний вираз функції (вираз можна скопіювати):

```
>> plot (t, 4-4*exp(-2*t))  
>> plot (t, 8*exp(-2*t))
```

Аналіз графіків динамічних і частотних характеристик

Розглянемо основні характерні точки характеристик, відображення яких на графіках задається з контекстного меню графіка (контекстне меню відкривається за допомогою правої кнопки миші). Числові значення цих точок можна довідатися, вказавши курсором відповідну точку на графіку й натиснувши ліву кнопку миші.

Step (перехідна характеристика.

- **Peak Response** - максимальне значення перехідного процесу.

- **Setting Time** - час регулювання - інтервал часу, протягом якого перехідний процес входить в інтервал $\pm a\%$ щодо сталого значення. Значення a задається параметром Show setting time within у вкладці Characteristics вікна LTI Viewer Preferences і за замовчуванням дорівнює 2%.
- **Rise Time** - час наростання - інтервал часу, протягом якого перехідний процес від значення $\beta\%$ перший раз досягає значення $\gamma\%$ від сталого значення. Значення β та γ задаються параметром Show rise time from у вкладці Characteristics вікна LTI Viewer Preferences і за замовчуванням рівні 10% й 90% відповідно.
- **Steady State** - стале значення перехідного процесу.

Impulse - імпульсна характеристика.

- **Peak Response** - максимальне значення імпульсної характеристики.
- **Setting Time** - час встановлення - інтервал часу, протягом якого закінчується перехідний процес.

Bode - логарифмічна амплітудна й фазова частотні характеристики.

- **Peak Response** - максимальне значення логарифмічної частотної характеристики.
- **Stability Margins** - запаси стійкості по амплітуді й фазі. Відображаються точки, у яких логарифмічна частотна характеристика дорівнює 0дБ й/або фазова характеристика дорівнює 180° .

Nyquist (амплітудно-фазова частотна характеристика (крива Найквіста).

- **Stability Margins** - запас стійкості. Відображається точка, у якій відстань від характеристики до точки $(-1; j\cdot 0)$ мінімальна.

Nichols - діаграма Нікольса.

По горизонтальній осі відкладається фаза в градусах, а по вертикальній - коефіцієнт передачі в дБ. Параметри графіка (задаються в меню Plot Options):

- **Stability Margins** - запас стійкості. Відображається точка, що характеризує запас стійкості.

Singular Value - особливе значення.

Будується логарифмічна амплітудно-частотна характеристика системи.

Pole-Zero Map - розташування нулів і полюсів.

Координати нуля або полюса можна довідатися, указавши курсором відповідну точку на графіку й натиснувши ліву кнопку миші.

Передатна функція електричного ланцюга

Передатну функцію електричного ланцюга RCL можна визначити на підставі складання диференціального рівняння. Цей спосіб громіздкий, тому застосовують метод операторних опорів, згідно якого відразу знаходять передатну функцію. Наприклад, необхідно знайти передатну функцію ланцюга:

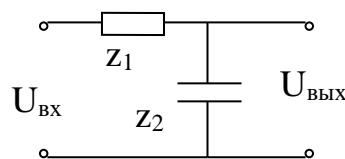


Рис. 7.14. Електричний ланцюг.

Операторний опір ланцюга зі сторони виходу при розімкнутому вході:

$$Z_{\text{ВЫХ}} = Z_2(s).$$

Операторное опір ланцюга з боку входу при розімкнутому виході:

$$Z_{\text{ВХ}} = Z_2(s) + Z_2(s).$$

$$W(s) = \frac{U_{\text{вих}}(s)}{U_{\text{вх}}(s)} = \frac{z_{\text{вих}}(s)}{z_{\text{вх}}(s)} = \frac{z_2(s)}{z_1(s) + z_2(s)}.$$

Робимо заміну згідно таблиці, у якій $s=j\omega$:

Параметр	R	L	C
$z(s)$	R	s	$\frac{1}{s}$

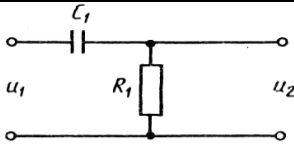
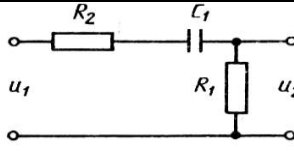
При цьому, постійна часу $T=RC$ або $T=\frac{R}{L}$.

Для конкретного електричного ланцюга одержимо:

$$W(s) = \frac{1/sC}{R + 1/sC} = \frac{1}{RCs + 1} = \frac{1}{Ts + 1}.$$

Таблиця 7.1. Варіанти завдання до лабораторної роботи №7.

Типові коригуючі пристрої систем автоматичного регулювання на постійному струмі

Група 1 коригуючих пристроїв					
Коригуючі пристрої з випередженням по фазі й найбільшим нахилом амплітудної характеристики 20дБ/дек					
1		2		3	
4		5		7	
7					

Коригуючі пристрої з випередженням по фазі й з найбільшим нахилом амплітудної характеристики 40 дБ/дек					
8		9			
Група 2 коригуючих пристроїв					
Коригуючі пристрої із запізнюванням по фазі й найбільшим нахилом амплітудної характеристики 40 дБ/дек					
1 і 2		3 і 4			
Коригуючі пристрої із запізнюванням по фазі й найбільшим нахилом амплітудної характеристики до 20 дБ/дек					
5 і 6		8		10	
7		9			

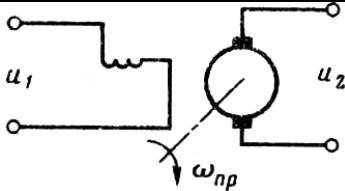
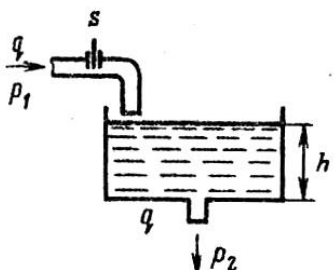
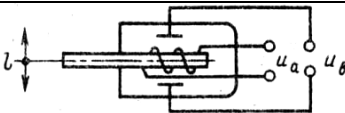
Таблиця 7.2. Варіанти завдання до лабораторної роботи №7.

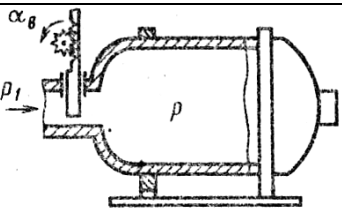
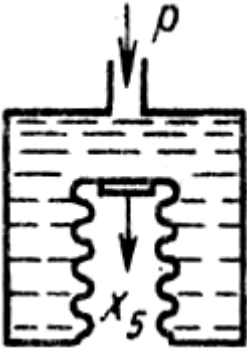
Передатні функції основних типових ланок

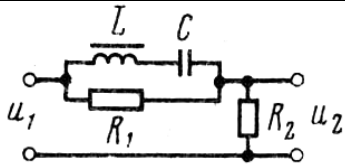
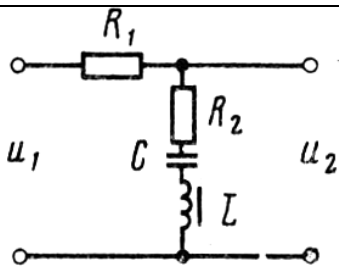
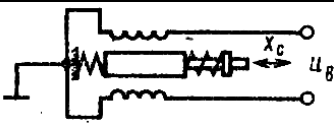
№	Найменування ланки	Передатна функція $W(s)$
1	Стійка аперіодична	$\frac{1}{Ts + 1}$
2 і 3	Нестійка аперіодична	$\frac{1}{Ts - 1}$ та $\frac{1}{1 - Ts}$

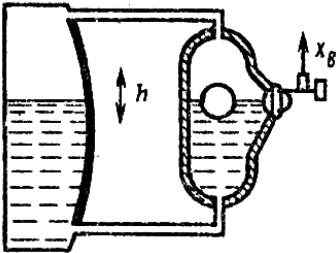
4	Інтегруюча n-го порядку	$\frac{1}{s^n}$, де $n=1, 2 \dots$
5 і 6 і 7	Диференціююча 1-го порядку	$Ts+1$ та $Ts-1$ та $1-Ts$
8	Стійка коливальна $\xi>0$	$\frac{1}{T^2s^2 + 2\xi Ts + 1}$
9	Нестійка коливальна $\xi<0$	$\frac{1}{T^2s^2 - 2\xi Ts + 1}$
10		$\frac{1}{T^2s^2 + 2\xi Ts - 1}$
11		$\frac{1}{T^2s^2 - 2\xi Ts - 1}$
12	Вирождена коливальна $\xi=0$	$\frac{1}{T^2s^2 + 1}$
13	Диференціююча 2-го порядку $\xi<0$	$T^2s^2 + 2\xi Ts + 1$
14		$T^2s^2 - 2\xi Ts + 1$
15		$T^2s^2 + 2\xi Ts - 1$
16		$T^2s^2 - 2\xi Ts - 1$
17	Вирождена диференціююча $\xi=0$	$T^2s^2 + 1$
18	Вирождена диференціююча n-го порядку	s^n , де $n=1, 2 \dots$
19	Чистого запізнювання	$e^{-\tau s}$
20	Трансцендентні	$e^{-\sqrt{\tau s}}$ та $-\text{th}(\tau s)$ та $-\frac{\text{th}(\tau s)}{1 + \text{th}(\tau s)}$
21	Підсилювальна	K

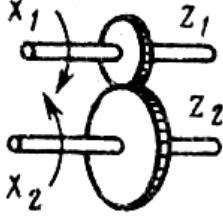
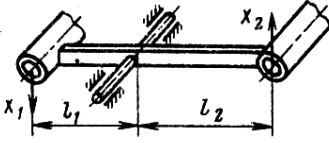
Таблиця 7.3. Варіанти завдання до лабораторної роботи №7.
Передатні функції елементів (об'єктів) систем автоматичного
регулювання

№	Найменування елемента або об'єкта системи	Схема, елемента або об'єкта	Передатна функція елемента або об'єкта.
1	Генератор		$W(s) = U_1(s)/U_2(s) =$ $= K_{\Gamma} / (T_{\Gamma} s + 1)$ <p>T_{Γ} – постійна часу обмотки збудження</p> <p>K_{Γ} – коефіцієнт підсилення генератора по напрузі</p>
2	Резервуар з рідиною		$W(s) = H(s)/S(s) = K / (TS + 1)$ $T = \frac{2(p_1 - \gamma h_0)(\gamma h_0 - p_2)}{\gamma(p_1 - p_2)} - \frac{1}{q}$ $K = \frac{2(p_1 - \gamma h_0)(\gamma h_0 - p_2)}{\gamma(p_1 - p_2)}$ <p>γ - питома вага рідини;</p> <p>h - максимальна висота підйому рідини</p>
3	Механотрон		$W(s) = U_B(s)/L(s) = K_M / (T_M s + 1)$ <p>K_M - коефіцієнт підсилення механотрона</p> <p>T_M – постійна часу механотрона</p>

4	Термопара		$W(s) = \vartheta(s)/U_b(s) = K_{ТП}/(T_{ТП}s + 1)$ $T_{ТП}$ - термопары; $K_{ТП}$ - коефіцієнт підсилення термопары
5	Ресивер		$W(s) = P_1(s)/\alpha_b(s) = K_p/(T_p s + 1)$ T_p - постійна часу ресивера K_p - передатний коефіцієнт ресивера
6	Мембранні перетворювачі з рідинним наповнювачем		$W(s) = X_K(s)/P(s) = K_K/(T_K s + 1)$ K_K - передатний коефіцієнт мембранного перетворювача T_K - постійна часу мембранного перетворювача
7	Тахогенератор		$W(s) = U_{ТГ}(s)/\beta(s) = K_{ТГ}s$ $K_{ТГ}$ - крутість характеристики тахогенератора
8	Міст RC		$W(s) = E_2(s)/E_1(s) = (Ts - 1)/(Ts + 1)$ $T = R_1 C_1 = R_2 C_2$;

9	Коригувальний пристрій змінного струму фазовипереджаючого типу		$W(s) = E_2(s) / E_1(s) =$ $= K T_1 s / (T_1 s + 1)$ $K = R_1 / (R_1 + R_2);$ $T_1 = L / R_1$
10	Коригувальний пристрій змінного струму запізнюючого типу		$W(s) = E_2(s) / E_1(s) =$ $= K (T_1 s + 1) / (T_2 s + 1)$ $K = R_1 / (R_1 + R_2);$ $T_1 = L / R_1;$ $T_2 = K T_1$
11	Соленоїд		$W(s) = \frac{X_c(s)}{U_B(s)}$ $= \frac{K_c}{T_c^2 s^2 + 2\xi_c T_c s + 1}$ <p> K_c – передатний коефіцієнт соленоїда; T_c – постійна часу соленоїда; ξ_c – коефіцієнт соленоїда </p>

12	Поплавковий рівнемір		$W(s) = \frac{X_B(s)}{H(s)}$ $= \frac{K_{\text{ПУ}}}{T^2 s^2 + 2\xi Ts + 1},$ <p>де $T = \sqrt{\frac{m}{K_c + \gamma \frac{\pi d^2}{4}}}$;</p> $K_{\text{ПУ}} = \frac{\gamma \frac{\pi d^2}{4}}{K_c + \gamma \frac{\pi d^2}{4}} ;$ $\xi = \frac{D}{2 \sqrt{m \left(K_c + \gamma \frac{\pi d^2}{4} \right)}} ;$ <p>m – маса поплавця D – коефіцієнт рідинного тертя; d – діаметр поплавця x – переміщення поплавця; h – зміна рівня рідини; γ – питома щільність рідини; K_c – коефіцієнт твердості пружини</p>
----	-------------------------	---	---

13	Механічна передача (редуктор)		$W(s) = \frac{X_2(s)}{X_1(s)}$ <p>x_1 і x_2 кути повороту валів</p> <p>$i_{ред} = \frac{z_2}{z_1}$ – передаточне число редуктора</p>
14	Важіль		$W(s) = \frac{X_2(s)}{X_1(s)} = \frac{1}{K_p}$ $K_p = \frac{L_2}{L_1}$

Контрольні запитання

1. Приведіть модель для одержання статичної характеристики динамічної системи шляхом моделювання за допомогою Simulink.
2. Приведіть модель для одержання перехідної характеристики динамічної системи шляхом моделювання за допомогою Simulink.
3. Приведіть модель, для одержання імпульсної характеристики динамічної системи шляхом моделювання за допомогою Simulink.
4. Поясніть яка методика побудови перехідної, імпульсної перехідної та частотних характеристик за допомогою вікна LTI Viewer Control System Toolbox.
5. Поясніть яка методика побудови перехідної, імпульсної перехідної та частотних характеристик за допомогою функцій Control System Toolbox,
6. Як отримати аналітичні вирази та будувати перехідну й імпульсну характеристики за допомогою Symbolic Math?
9. Поясніть призначення блоку TransferFcn, які параметри цього блоку?

7. Які сигнали потрібно подавати на вхід блоку TransferFcn для координат точок частотних характеристик динамічної системи?
8. Поясніть, залежність між якими величинами відображає статична характеристика.
9. Поясніть, залежність між якими величинами відображає перехідна характеристика.
10. Поясніть, залежність між якими величинами відображає імпульсна перехідна (вагова) характеристика.
11. Поясніть, залежність між якими величинами відображає амплітудна частотна характеристика.
12. Поясніть, залежність між якими величинами відображає фазова частотна характеристика.
13. Поясніть, залежність між якими величинами відображає амплітудно-фазова характеристика.
14. Поясніть, залежність між якими величинами відображає логарифмічна амплітудна частотна характеристика.
15. Поясніть, залежність між якими величинами відображає логарифмічна фазова частотна характеристика.
16. Як отримати передатну функцію коригувального пристрою.

ПЕРЕЛІК БЛОКІВ SIMULINK

1. Розділ Sources (джерела)

Блоки розділяють Sources - джерела сигналів (вхідних величин):

1.1. Band-Limited White Noise — джерело білого частотно-обмеженого шуму.

Параметри налаштування блоку:

- Noise power - потужність білого шуму;
- Sample time — дискрета часу (визначає верхнє значення частоти процесу);
- Seed - початкове значення бази генератора випадкової величини

1.2. Chirp Signal — джерело синусоїдального сигналу одиничної амплітуди й змінної частоти, частота змінюється в часі за лінійним законом.

Параметри налаштування блоку:

- Initial Frequency(Hz) - початкова частота, Гц;
- Targeted time - наступний (>0) момент часу, з;
- Frequency at target time - частота в цей наступний момент часу, Гц .

1.3. Constant— джерело постійної величини (скаляра, вектора або матриці).

Параметри налаштування блоку:

- Constant value — значення константи, наприклад, 4—число;
[1 4 7]—вектор; [1 4 7][-5 pi 1e-5]—матриця; $\sin(3) + 2$ — обчислює враз.

Може використатися як джерело випадкових чисел. Для цього потрібно звернутися до генератора безперервної випадкової величини, рівномірно розподіленої в заданому інтервалі: `unifrnd (A,B,M,N)`, де A,B — границі

діапазону розподілу, M, N задають розмір матриці випадкових чисел, яка буде згенерована.

1.4. Clock— генератор безперервного сигналу часу. Його робота заснована на використанні апаратного таймера комп'ютера

Параметри налаштування блоку:

- Display time (виводити/не виводити час у пітограмму блоку;
- Decimation (крок генерування вихідного сигналу дорівнює добутку цього параметра на крок зміни модельного часу, заданому в параметрах Simulation parameters.

1.5. Digital clock— джерело дискретного сигналу часу, тобто час моделі буде змінюватися тільки в задані моменти часу.

Параметри налаштування блоку:

- Sample time - крок генерування сигналу даним блоком, якщо встановити (-1), то крок буде дорівнювати кроку зміни модельного часу, заданому в параметрах Simulation parameters.

1.6. From File — введення в блок-модель даних, що зберігаються в mat-файлі.

Параметри налаштування блоку:

- File name - ім'я mat-файлу, з якого будуть зчитуватися дані;
- Sample time - періодичність зчитування даних.

mat-файл, з якого зчитуються дані, повинен містити матрицю, що складається із двох або більше рядків. Перший рядок повинен містити монотонно зростаючі значення модельного часу, інші рядки - дані, що відповідають цим значенням часу, як показано нижче:

Перший рядок	$t_1,$	$t_2,$	t_3	...	t_{jn}
Другий рядок	$u_{11},$	$u_{12},$	u_{13}	...	u_{1fin}
n-й рядок	u_{n1}	u_{n2}	u_{n3}	...	u_{nfin}

Якщо використовуваний при моделюванні крок часу не збігається зі значеннями, записаними в першому рядку файлу, то SIMULINK виконує лінійну інтерполяцію даних; аналогічно, якщо перше значення (t_1) менше часу початку моделювання, або останнє значення (t_{fin}) більше часу закінчення моделювання, то SIMULINK виконує екстраполяцію даних.

У розділі Sinks є блок To File, що виконує запис результатів моделювання в mat-файл. Однак варто мати на увазі, що SIMULINK завантажує дані з файлу один раз при запуску моделі; тому неможливо в тому самому сеансі моделювання й записувати дані в mat-файл, і зчитувати їх з нього.

1.7. From Workspace — уведення в модель даних з робочої області MATLAB.
--

Workspace (робоча область - це власна оперативна пам'ять MATLAB.

Параметри налаштування блоку:

- Data –дані вводяться в Workspace з командного рядка MATLAB.

Наприклад, у вигляді двох матриць, де значення **Data** буде – [T,U], у такий спосіб:

>>T=[0:0.1:10]’ – вектор значень моментів часу, тобто час міняється від 0 до 10 із кроком 0.1; символ ’ – вказує завдання вектора-стовпця

>>U=[sin(T) cos(T)] – матриця значень вихідних сигналів у відповідні моменти часу.

Або, наприклад, у вигляді однієї матриці , тоді значення **Data** буде – [U] у такий спосіб:

>>U=[0:0.1:10; sin(0:0.1:10)]’ – одна матриця, 1-ий стовпчик якої містить значення часу, а інші - значення сигналів.

- Sample time - періодичність зчитування даних;

- Interpolate data - інтерполювати дані, якщо задані моменти часу не збігаються з необхідними.

1.8. Ground (земля) і **Terminator** (обмежник) можуть використовуватися як заглушки для тих портів, які з якої-небудь причини виявилися не підключеними до інших блоків моделі. Причому Ground використовується для вхідних портів, а Terminator - для вихідних портів.

1.9. In (вхідний порт) вхідний порт підсистеми, разом з **Out** (вихідний порт) забезпечують зв'язок між підсистемами моделі за інформацією.

Параметри налаштування блоку:

- Port number - номер порту;
- Port width - кількість векторів вхідного порту (-1 для вектора з динамічною розмірністю);
- Sample time - період дискретизації (-1 для безперервних систем);
- Data type - тип вхідних даних;
- Signal type - тип сигналу:
 - auto - тип визначається автоматично;
 - real - дійсний;
 - complex - комплексний.
- Interpolate data - інтерполювати дані, якщо значення моментів часу сигналів, що вводять через порт, не збігаються з необхідними

1.10. Pulse Generator — генератор імпульсних сигналів прямокутної форми.

Параметри налаштування блоку:

- Amplitude — амплітуда (висота) імпульсу;
- Period— період проходження імпульсів, с;
- Pulse Width(% of period) — тривалість імпульсу в % від періоду;

- Phase delay — час початку генерування імпульсів, с.

1.11. Ramp — генератор лінійно зростаючого або убуючого сигналу.

Параметри налаштування блоку:

- Slope — крутість нахилу прямиї до осі часу;
- Start time — час до початку наростання сигналу;
- Initial output — початкове значення сигналу.

1.12. Random Number — джерело дискретного сигналу, амплітуда якого є випадковою величиною, розподіленою за нормальним законом.

Параметри налаштування блоку:

- Mean — середнє значення;
- Variance — дисперсія, тобто середнє квадратичне відхилення від середнього;
- Initial seed — початковий параметр генерації, який необхідно змінювати для одержання нової послідовності випадкових чисел;
- Sample time - крок генерування сигналу даним блоком, якщо встановити (-1), то крок буде дорівнює кроку зміни модельного часу, заданому в параметрах Simulation parameters.

1.13. Repeating Sequence — генератор періодичного дискретного сигналу, форма якого задається в параметрах блоку і являє собою ламану лінію, що проходить через точки із зазначеними у векторах Time values й Output values координатами.

Параметри налаштування блоку:

- Time values — значення моментів часу, наприклад, [0 1 1 2];
- Output values — значення вихідного сигналу в ці моменти часу, наприклад, [0 1 2 1].

1.14. Signal Generator — генератор безперервного сигналу довільної форми.

Параметри налаштування блоку:

- Wave form - форма періодичного процесу: sine (синусоїдальні коливання), square (прямокутні коливання), sawtooth (трикутні коливання) або random (випадкові коливання);
- Amplitude - амплітуда коливань, які генеруються блоком;
- Frequency - частота коливань;
- Units - одиниця виміру частоти Hertz (Герци) або Rad/Sec (радіан/с).

1.15. Step — генератор типового впливу у вигляді одиничної сходинки.

Параметри налаштування блоку:

- Step time — час генерування початкового значення сигналу;
- Initial value — початкове значення сигналу;
- Final value — кінцеве значення сигналу;
- Sample time — крок генерування сигналу даним блоком, якщо встановити (–1), то крок буде дорівнювати кроку зміни модельного часу, заданому в параметрах Simulation parameters.

1.16. Sine Wave — генератор гармонійних коливань.

Параметри налаштування блоку:

- Amplitude— амплітуда синусоїдального сигналу;
- Bias— переміщення синусоїди по осі ординат;
- Frequency(rad/sec) — частота коливань, радіан/с;
- Phase(rad) — початкова фаза, радіан;
- Sample time — крок генерування сигналу даним блоком, якщо встановити (–1), то крок буде дорівнювати кроку зміни модельного часу, заданому в параметрах Simulation parameters.

1.17. Uniform Random Number — джерело дискретного сигналу, амплітуда якого є рівномірно розподіленою випадковою величиною.

Параметри налаштування блоку:

- Minimum - нижня границя випадкових чисел;
- Maximum - верхня границя;
- Initial seed — початкове значення бази генератора випадкових чисел. При фіксованому значенні цього параметра генератор завжди виробляє ту саму послідовність випадкових чисел;
- Sample time - дискрет часу.

Особливістю блоку є те, що він у кожному сеансі моделювання генерує ту саму послідовність випадкових чисел. Для зміни послідовності, що генерується, необхідно змінити значення параметра Initial seed (початкове значення).

2. Розділ Sinks (одержувачі)

Блоки, використовувані при моделюванні у якості «проглядових вікон»:

2.1. Scope – осцилограф (індикатор).

Дозволяє в процесі моделювання отримати графіки зміни любых сигналів блок-моделі. По горизонтальній осі відкладаються значення модельного часу, а по вертикальній - значення вхідної величини. Якщо вхідна величина є вектором, у вікні виводяться графіки всіх елементів цього вектора (дозволяється до 30 кривих).

Для налаштування параметрів блоку потрібно двічі клацнути на його зображенні для появи вікна Scope. Панель інструментів вікна Scope зображено на рис.1.

Натискання піктограми Parameters (параметри) відкриває вікно налаштування параметрів Scope Parameters із двома вкладками General — параметри керування форматом зображення графіків; Data history (протоколювання даних) — параметри запису в робочу область MATLAB відображуваних на графіках даних.

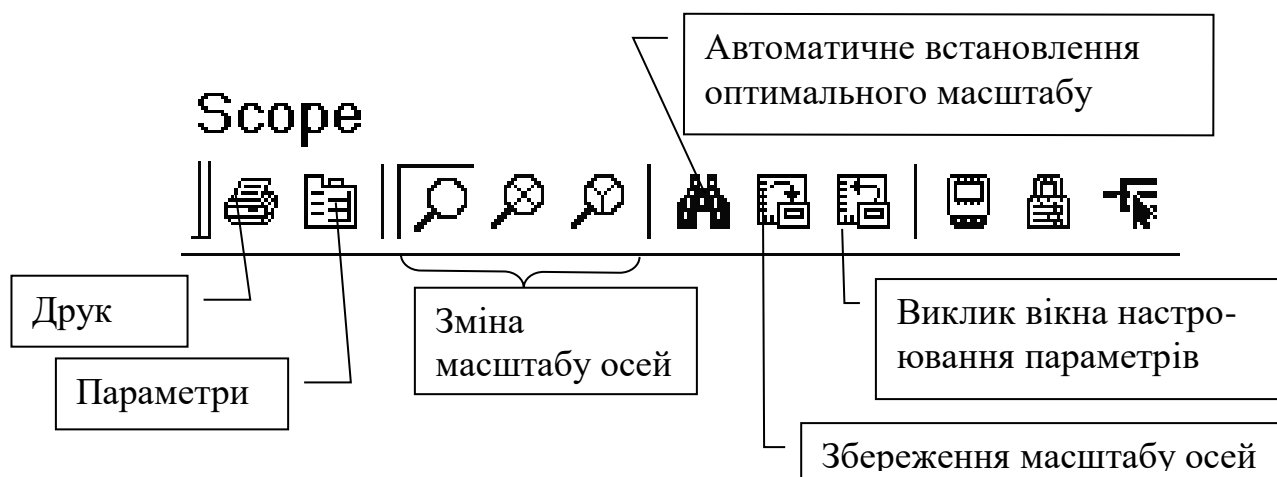


Рис. 1. Панель інструментів вікна Score.

Вкладка General містить наступні елементи:

- Number of axes - число підвікон і число вхідних портів блоку Score;
- Time range - граничне значення діапазону по осі часу, у разі, якщо значення параметру “auto”, то граничне значення часу збігається з кінцевим значенням модельного часу, установленого для сеансу моделювання);
- Tick labels (тактові мітки) - дозволяє вибрати формат подання осі X: bottom axis only (значення часу виводяться тільки для нижнього графіка); all (на всіх); none (не виводяться);
- Sampling (періодичність) призначений для вибору варіанта керування періодичністю малювання графіків: Decimation (дискретність) - коефіцієнт «проріджування» виведених значень, наприклад, якщо задати для цього параметра значення 3, то спостережувальні характеристики відображаються у вікні Score тільки на кожному третьому кроці моделювання; Sample time (еталонний час) - періодичність визначається величиною кроку модельного часу, установленого для сеансу моделювання;

- floating scope – дозволяє визначати блок, як блок без входу, обриваючи його зв'язки. Блок буде відображати сигнал, який проходить по виділеній в даний момент часу лінії зв'язку.

Вкладка Data history містить наступні елементи:

- Limit rows to last (граничне число рядків) - об'єм даних, що зберігають;
- Save data to workspace спосіб запису, якщо він установлений, то відображувані у вікні Scope дані зберігаються в робочій області MATLAB у вигляді матриці або структури;
- Variable Name - ім'я матриці або структури для зберігатися даних;
- Save format дозволяє вибрати формат запису: Matrix (у вигляді матриці); Structure (у вигляді структури); Structure With Time (у вигляді структури зі значеннями модельного часу). Структура - це тип даних, використовуваний у багатьох мовах програмування високого рівня. Від матриці або масиву структура відрізняється тим, що її елементи (поля) можуть бути різнотипними (цілими, дійсними, символьними й т.д.).

При збереженні даних у форматі Matrix у робочій області створюється масив значень, у якому число стовпців дорівнює числу спостережуваних параметрів, а число рядків залежить від періодичності реєстрації параметрів і тривалості сеансу моделювання; їхня максимальна кількість визначається параметром Maximum number of rows.

При виборі формату Structure дані зберігаються у вигляді структури, що містить три поля: time (вектор значень модельного часу), signals (масив значень спостережуваних параметрів) і blockName (ім'я блоку To Workspace, використовуване в моделі); вектор time для цього формату не заповнюється.

Формат Structure With Time відрізняється від попередніх тим, що забезпечує заповнення вектора time.

Вибір формату збереження даних визначається тим, яким чином ці дані будуть згодом використовуватися. Наприклад, якщо вони повинні будуть завантажуватися в модель за допомогою блоку From File, то обов'язково потрібно зберегти вектор значень модельного часу.

Деякі налаштування блоку можуть бути також виконані за допомогою команд контекстного меню, що викликається на екран натисканням правою кнопкою миші (показчик при цьому повинен перебувати усередині координатної сітки)

2.2. XYGraph (двовимірний графік) – створення двовимірних графіків залежностей довільних величин, що фігурують у моделі, у прямокутній системі координат.

Блок має два входи, перший з яких призначений для введення аргументу, другий - для введення значень функції цього аргументу.

Параметри налаштування блоку:

- X-min, X-max, Y-min, Y-max - діапазони осей графіка;
- Sample time - крок модельного часу, якщо дорівнює -1, то величина кроку визначається значенням однойменного параметра блоків, з яких надходять вхідні сигнали на блок XYGraph.

2.3. Display (екран) – відображення чисельних значень величин.

Параметри налаштування блоку:

- Format задає формат виводу чисел: short, long, short_e, long_e, bank;
- Decimation дозволяє задати періодичність виводу значень у вікні Display;
- Floating display дозволяє визначати блок Display як блок без входу, обриваючи його зв'язки.

Якщо відображувана величина є вектором, то з'являється чорний трикутник у правому нижньому куті блоку. Для всіх з елементів вектора

створюються свої міні-вікна, для перегляду яких, необхідно розтягти зображення блоку.

Блоки, що забезпечують збереження результатів моделювання:

2.4. To File – забезпечує запис значень величини, поданої на його вхід, в mat-файл.

Параметри настроювання блоку:

- File name (ім'я файлу) - ім'я mat -файлу для запису даних;
- Variable name - ім'я змінної для звертання до даних, записаних у файлі, за замовчуванням це ім'я “ans”.
- Decimation - дискретність запису даних у файл. При введенні значення (-1) запис відбувається на кожному кроці моделювання.
- Sample Time - величина кроку моделювання для даного блоку.

Дані з файлу для перегляду й редагування можна завантажити в робочу область за допомогою команди File>Load Works, наприклад:

```
>> ans
```

```
ans =
```

```
0 } 1 }  
1 } 5 }  
2 } 3 }
```

Моменти часу

Значення вихідного сигналу

2.5. To Workspace – запис даних у робочу область MatLAB.

Параметри настроювання блоку:

- Variable name - ім'я змінної, під яким дані зберігаються в робочій області (за замовчуванням - simout).
- Limit Data Points to last (максимальне число рядків) — припустиме число кроків моделювання, для якого реєструються дані (за

замовчуванням `inf`, тобто дані реєструються на всьому інтервалі моделювання).

- **Decimation** - дискретність реєстрації даних.
- **Sample time** - крок генерування сигналу блоком, якщо значення рівне (-1), то крок буде дорівнювати кроку зміни модельного часу, заданому в параметрах **Simulation parameters**.
- **Save format** - формат запису даних:
 - **Array** - у вигляді матриці, у якій число стовпців дорівнює числу спостережуваних сигналів, а число рядків – кількості прорахунків у процесі моделювання;
 - **Structure** - у вигляді структури, що містить поля: `time` (модельний час), `signals` (значення спостережуваних сигналів), `blockName` (ім'я блоку **To Workspace**); `time` для цього формату не заповнюється;
 - **Structure With Time** - відрізняється від попередніх тим, що забезпечує заповнення вектора `time`

Формат даних визначається тим, яким чином ці дані будуть згодом використовуватися.

Для перегляду переліку даних робочої області існує вікно перегляду, що викликається **View>Workspace**. У робочій області присутній вектор моментів часу реєстрації - `tout`, якщо на вкладці **Workspace I/O** (обмін з робочою областю) вікна параметрів моделювання встановлений прапорець **Time**. Кнопка **Open** дозволяє завантажити дані в редактор **MATLAB** у вигляді електронної таблиці, уміст якої може коректуватися й знову зберігатися в робочій області. Дані з робочої області можна зберегти в `mat`-файлі командою **File >Save Workspace As**.

Для перегляду вмісту даних у командному вікні MATLAB наберіть ім'я змінної й Enter. Наприклад, для змінної `sim`:

```
>> sim
```

```
sim =
```

```
1    1    0
0    0    1
```

Матриця `Sim` складається зі значень 3 сигналів, зареєстрованих 2 рази: на першому кроці моделювання, коли модельний час дорівнює 0, і на другому кроці.

Блоки керування моделюванням.

2.6. Stop Simulation – зупинка моделювання, при надходженні на вхід блоку ненульового сигналу. Не має параметрів настроювання.

2.7. Terminator (обмежник) і **Ground** (земля) можуть використовуватись у якості заглушок для тих портів, які з якої-небудь причини виявилися не підключеними до інших блоків моделі. Причому **Ground** використовується для вхідних портів, а **Terminator** - для вихідних портів.

2.8. Out (вихідний порт) – вихідний порт підсистеми, разом з **In** (вхідний порт) забезпечують зв'язок між підсистемами моделі за інформацією.

Параметри настроювання блоку:

- **Port number** - номер порту
- **Output when disabled** - значення виходу, якщо підсистема, що містить порт, заборонена
- **Initial output** - початкове значення виходу. Задає початкове значення для умовно виконуваних систем, якщо система заборонена

3. Розділ Nonlinear (нелінійні системи)

Блоки розділу, які виконують функції блоків-обмежників:

3.1. Dead Zone (пасивна зона) заміняє значення вхідного сигналу, що лежить в заданому діапазоні, нулем, а значення, яке не лежить в заданому діапазоні наближає до границь цього діапазону, змінюючи їх.

- Start of dead zone (початок пасивної зони) -нижня границя діапазону;
- End of dead zone (кінець пасивної зони) -верхня границя діапазону.

3.2. Backlash (люфт) – забезпечує «утримання» вхідного сигналу в деякому заданому діапазоні.

- Deadband width (ширина діапазону).
- Initial output (початкове значення вихідного сигналу) - відіграє роль середини діапазону; наприклад, якщо Deadband width=2, Initial output=1, те нижня границя діапазону встановлюється рівною 2, а верхня - рівною 0.

3.3. Rate Limiter (обмежник швидкості) – для контролю й обмеження швидкості зміни (тобто першої похідної) вхідного сигналу.

Значення похідної (rate) визначається з наступного співвідношення:

$$\text{rate} = [U(i) - \text{OUT}(i-1)] / [T(i) - T(i-1)],$$

де $U(i)$ і $T(i)$ - поточні значення вхідного сигналу й модельного часу, а $\text{OUT}(i-1)$ і $T(i-1)$ - значення на попередньому кроці моделювання.

Значення вихідного сигналу визначається на основі порівняння величини rate з її граничними значеннями, заданими у вікні настроювання блоку.

- Якщо rate більше параметра R, заданого в полі Rising slew rate (збільшення швидкості), то вихідний сигнал обчислюється так:

$$\text{OUT}(i) = (T-R + \text{OUT}(i-1)).$$

- Якщо rate менше параметра F, заданого в полі Falling slew rate (зниження швидкості), то вихідний сигнал обчислюється так:

$$OUT(i) = (T-F + OUT(i-1)).$$

- Якщо rate перебуває в проміжку між значеннями параметрів Rising slew rate й Falling slew rate, то вихідний сигнал дорівнює вхідному:

$$OUT(i) = U(i).$$

3.4. Saturation (насичення) — вихідна величина цього блоку збігається із вхідною, якщо остання перебуває усередині зазначеного діапазону. Якщо ж вхідна величина виходить за рамки діапазону, то вихідний сигнал приймає значення найближчої із границь. Значення границь діапазону встановлюються у вікні налаштування блоку.

Блоки, що управляють напрямком передачі сигналу - «перемикачі»:

3.5. Switch – перемикає до виходу один із двох інформаційних входів залежно від сигналу на керуючому вході.

Має два (1-й і 3-й) інформаційних та один (2-й) – керуючий входи. Якщо величина керуючого сигналу більше чи дорівнює значенню Threshold (поріг), то на вихід блоку передається сигнал з 1-го входу, інакше - сигнал з 3-го входу.

3.6. Manual Switch – ручний перемикач.

Дозволяє вручну перемкнути з'єднання одного із двох входів до виходу подвійним натисканням мишею на зображенні блоку.

3.7. Multiport Switch – багатовходовий перемикач.

Має не менше трьох входів. Перший зверху є керуючим, інші - інформаційними. Параметр налаштування Number of inputs встановлює число інформаційних входів. Номер входу, що з'єднується з виходом, дорівнює значенню керуючого сигналу, що надходить на верхній вхід. Якщо це значення є дробовим числом, то округляється до цілого за

звичайними правилами, якщо менше одиниці, вважається рівним 1, а якщо більше числа інформаційних входів, то приймається рівним найбільшому номеру (входи нумеруються зверху вниз, крім самого верхнього - керуючого).

3.8. Relay працює аналогічно реле: якщо вхідний сигнал перевищує граничне значення, то на виході формується «дозволяючий» сигнал.

Параметри налаштування блоку:

- Switch on point (точка включення) - граничне значення, при перевищенні якого відбувається «включення» реле;
- Switch off point (точка вимикання) - рівень сигналу, при якому реле «вимикається»;
- Output when on (вихід при включеному стані) - значення амплітуди «дозволяючого» сигналу;
- Output when off (вихід при виключеному стані) - рівень сигналу на виході реле, коли воно перебуває в стані «виключено».

3.9. Quantizer (квантувач) здійснює дискретизацію вхідного сигналу по його величині.

Параметр налаштування блоку Quantization interval (Інтервал квантування) - величина дискрети за рівнем вхідного сигналу.

3.10. Coulomb&Viscous Friction (Сухе й вязке тертя) реалізує нелінійну залежність типу лінійна з попереднім натягом.

Якщо вхід позитивний, то вихід пропорційний входу й збільшений на величину натягу (кулонове тертя). Якщо вхід негативний, то вихід також пропорційний входу (з тим же коефіцієнтом пропорційності - коефіцієнтом вязкого тертя) за винятком величини натягу. При вході, рівному нулю, вихід також дорівнює нулю. У параметри налаштування блоку входять величини кулонового тертя (натягу) і коефіцієнта вязкого тертя.

4. Розділ Continuous (неперервні системи)

Блоки призначені для опису неперервних систем:

4.1. Transfer Fcn реалізує передатну функцію виду $W(s)=P(s)/Q(s)$.

Параметри настроювання блоку:

Numerator - чисельник, наприклад, [1 2 3];

Denominator - знаменник, наприклад, [4 5 6], тоді передатна функція буде мати вигляд: $W(s)=(s^2+2s+3)/(4s^2+5s+6)$.

Порядок знаменника повинен бути рівним чи більшим порядку чисельника.

4.2. State-Space (простір станів) — блок формування стану системи.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

- A, B, C, D - значення матриць A, B, C, D.
- Initial conditions - початковий стан змінних стану.

4.3. Zero-Pole (область переходу) — блок, що реалізує передатну функцію, описану за допомогою оператора Лапласа у вигляді полюсів, нулів й коефіцієнта передачі.

Параметри настроювання блоку:

- Zeros - вектор нулів, наприклад [1 2];
- Poles - вектор полюсів, наприклад [0 0 -1];
- Gain - коефіцієнт передачі, наприклад [3].

Передатна функція буде мати вигляд: $[3(s-1)(s-2)] / [s^2(s+1)]$.

4.4. Derivative (похідна) — блок обчислення похідної вхідного сигналу за часом (du/dt).

Блоки які, можуть бути використані в моделі будь-якої системи:

4.5. Memory (регістр) -затримка сигналу на один крок модельного часу.

Параметри налаштування блоку:

- **Initial condition** (початковий стан) - визначає значення амплітуди вхідного сигналу на момент ініціалізації блоку. У більшості випадків доцільно приймати це значення рівним нулю.
- **Inherit sample time** (успадковувати еталонний час) дозволяє задати величину кроку, на який буде виконуватись затримка сигналу:
 - якщо прапорець видалити, то буде використовуватись мінімальний крок, рівний 0.1 одиниці модельного часу;
 - якщо прапорець встановлений, то величина кроку визначається значенням у полі **Sample time** (еталонний час) блоку, що передує блоку **Memory**.

4.6. Transport Delay (затримка передачі) — забезпечує затримку сигналу на задану кількість кроків модельного часу, причому не обов'язково ціле.

Параметри налаштування блоку:

- **Time delay** (час затримки) - кількість кроків модельного часу, на яку затримується сигнал.
- **Initial input** (початкове значення на вході) - значення амплітуди вхідного сигналу в момент ініціалізації блоку (за замовчуванням дорівнює 0).
- **Initial buffer size** (початковий розмір буфера) - об'єм пам'яті (у байтах), виділеної в робочій області MATLAB для зберігання затриманого сигналу. Значення параметра повинно бути кратне 8 (за замовчуванням - 1024).

4.7. Variable Transport Delay (плаваюча затримка передачі) — блок, що реалізує динамічно змінювану затримку вхідного сигналу.

Блок дозволяє задавати керувану ззовні величину затримки. Для цього блок має додатковий вхід. Сигнал, який подається на блок ззовні, визначає тривалість затримки інформаційного сигналу, що надходить на основний

вхід блоку.

Параметри налаштування блоку:

- Maximum delay (максимальна затримка) - максимальний інтервал часу, на який може бути затриманий вхідний сигнал. Якщо величина затримки, обумовлена керуючим сигналом, перевищує це значення, то затримка приймається рівною значенню в полі Maximum delay.
- Initial input (початкове значення на вході) - значення амплітуди вхідного сигналу в момент ініціалізації блоку (за замовчуванням дорівнює 0);
- Buffer size (розмір буфера) - об'єм пам'яті (у байтах), яка виділяється в робочій області MATLAB для зберігання затриманого сигналу; значення параметра повинно бути кратне 8 (за замовчуванням - 1024).

4.8. Integrator — ідеальна інтегруюча ланка, здійснює інтегрування в безперервному часі вхідної величини

Робить обчислення «часу існування» вхідного сигналу й може бути використаний для визначення динамічних характеристик системи.

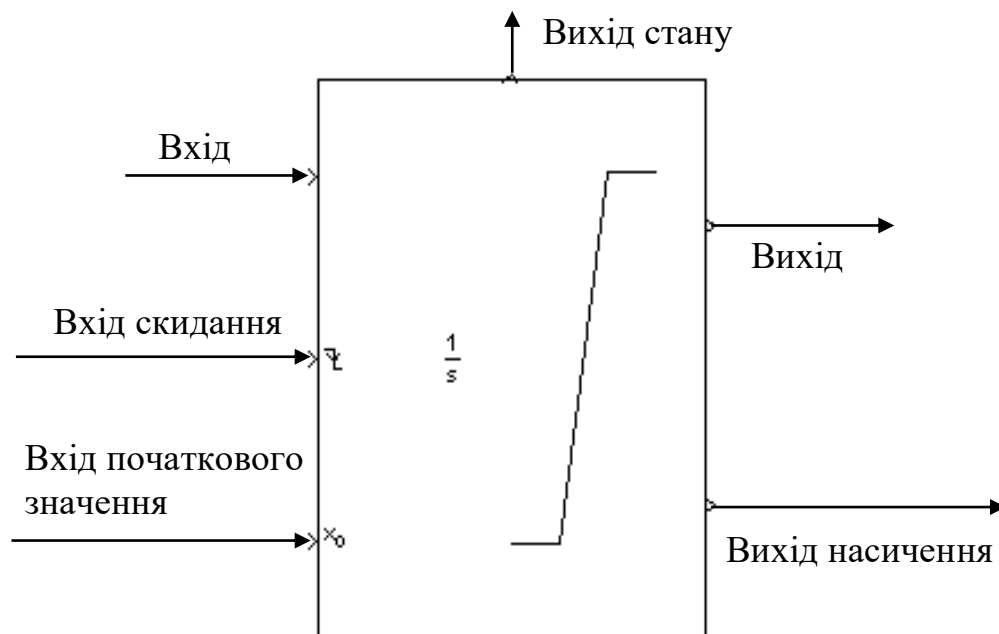


Рис.2. Входи й виходи інтегруючого блоку

Призначення входів і виходів:

- Вхід – вхід, на який подається сигнал, який буде інтегруватися блоком.
- Вхід скидання - передній фронт, задній фронт або обидва фронти імпульсу, який подається сюди, встановлюють інтегратор у початковий стан. Цей вхід включається установкою значення rising, falling або either у полі External reset.
- Вхід початкового значення - із цього входу при скиданні встановлюється початкове значення інтегратора. Цей вхід включається установкою значення external у полі Initial condition state.
- Вихід – вихід, де отримується результат інтегрування
- Вихід насичення - якщо інтегратор ненасичений, значення виходу дорівнює 0. Якщо інтегратор насичений, то значення виходу дорівнює величині насичення. Цей вихід включається полем Show saturation port.
- Вихід стану - спеціальний вихід, призначений для розриву алгебраїчних циклів у системі. Включається полем Show state port.

Наявність або відсутність деяких входів або виходів визначається завданням відповідних параметрів блоку. Вхід і вихід присутній при будь-яких значеннях параметрів.

Параметри налаштування блоку:

- External reset (зовнішній керуючий сигнал) дозволяє вибрати спосіб керування:
 - none - додатковий керуючий сигнал не використовується;
 - rising (підйом) - для керування використовується зростаючий сигнал;
 - falling (спад) - для керування використовується спадаючий сигнал;
 - either (кожної) - на роботу блоку впливає будь-яка зміна амплітуди керуючого сигналу.

Block Parameters: Integrator

Integrator

Continuous-time integration of the input signal.

Parameters

External reset:

Initial condition source:

☒ Limit output

Upper saturation limit:

Lower saturation limit:

☐ Show saturation port

☒ Show state port

Absolute tolerance:

none
rising
falling
either
level

internal
external

Рис. 3. Вікно налаштування блоку Integrator

- Initial condition source (джерело початкового стану) дозволяє вибрати одне із двох значень:
 - internal (внутрішня) - використається власна установка початкового значення суматора;
 - external (зовнішня) - установка початкового значення виконується ззовні.
- Initial condition - значення початкового стану;
- Limit output - обмеження вихідного значення;
- Upper saturation limit (верхнє граничне значення), за замовчуванням значення параметра не обмежене (inf);
- Lower saturation limit (нижнє граничне значення), за замовчуванням значення параметра не обмежене (-inf);

- Show saturation port (показати порт насичення);
- Show state port (показати порт стану);
- Absolute tolerance (точність обчислень).

Параметри, що залежать від прапорця Limit output, використовуються в такий спосіб. Якщо прапорець установлений, то при переході значення суматора через верхню або нижню межу на додаткових виходах блоку (портах насичення й стану) формується одиничний сигнал. Щоб цей сигнал можна було використати для керування роботою моделі, прапорці Show saturation port й Show state port повинні бути встановлені. Особливість порту стану полягає в тому, що сигнал, який в ньому виробляється, може бути використаний тільки для переривання алгебраїчного циклу або для узгодження стану підсистем моделі.

5. Розділ Math (математика)

Блоки реалізуючі математичні функції:

5.1. Abs формує абсолютне значення вхідного сигналу.

5.2. Trigonometric забезпечує перетворення вхідного сигналу за допомогою однієї з функцій: sin, cos, tan, asm, acos, atan, sinh, cosh, tanh.

5.3. Math Function задає для перетворення сигналу елементарні не тригонометричні й не гіперболічні функції, такі як обчислення експоненти, натурального й десяткового логарифмів, піднесення в ступінь, добування квадратного кореня й т.д.

5.4. Rounding Function містить різні функції округлення.

Метод округлення вибирається за допомогою списку у вікні налаштування.

5.5. MinMax здійснює пошук мінімального або максимального елемента вхідного вектора.

5.6. Complex to Magnitude-Angle (виділення з комплексного числа амплітуди й фази) дозволяє одержати із вхідної комплексної величини амплітудно-фазові характеристики сигналу.

5.7. Magnitude-Angle to Complex (перетворення амплітуди й фази в комплексне число).

5.8. Complex to Real-Image (виділення мнімої або дійсної частини комплексного числа) забезпечує виділення із вхідного комплексного сигналу його мнімої або дійсної частини (залежно від налаштування параметрів блоку).

5.9. Real-Image to Complex (створення комплексного числа з його мнімої й дійсної частини).

5.10. Product виконує множення або ділення декількох вхідних сигналів.

У параметри налаштування входять число входів блоку і вид виконуваної операції.

Параметри налаштування блоку:

- Number of inputs можна задати одним з наступних способів:
 - у вигляді послідовності знаків «*» або «/», що регламентують кількість входів блоку (якщо входів багато, їх доцільно розбивати на групи символом | (наприклад: **/*/*));

- задати кількість входів цифрою, на виході одержимо добуток вхідних сигналів, наприклад, введення константи 4 аналогічно введенню ****;
- цифра 1 указує, що буде відбуватися перемножування елементів вектора, що надходить на вхід блоку.
- Saturate on integer overflow обробляти переповнення для цілих, якщо результат перевищує значення припустиме для цілих.

5.11. Sum — ланка-суматор.

Може використатися у двох режимах: додавання вхідних сигналів; підсумовування елементів вектора, що надходить на вхід блоку.

Параметри настроювання блоку:

- List of signs (список знаків) можна задати одним з наступних способів:
 - у вигляді послідовності знаків «+» або «-», що регламентують кількість входів блоку (якщо входів багато, їх доцільно розбивати на групи символом | (наприклад: ++|--));
 - задати кількість позитивних входів цифрою, наприклад, введення константи 4 аналогічний введенню ++++.
 - цифра 1 визначає підсумовування елементів вектора, що надходить на вхід блоку (у цьому випадку усередині блоку виводиться символ ?).
- Saturate on integer overflow обробляти переповнення для цілих, якщо результат перевищує значення припустиме для цілих.

5.12. Dot product (згортка) – скалярне множення двох векторів, наприклад, згортка вектора $\begin{bmatrix} 2 & 3 \end{bmatrix}$ й $\begin{bmatrix} 1 & 4 \end{bmatrix}$ дорівнює 14.

Блоки, що забезпечують логічну обробку вхідного сигналу.

Вихідна величина в усіх цих блоках є булевою величиною, тобто може приймати значення: 1 ("істина") або 0 ("неправда").

5.13. Relational Operator реалізує операції відношення над двома вхідними сигналами.

Конкретна операція $>$, $<$, \geq , \leq , $==$ (тотожно), $!=$ (не дорівнює) вибирається при настроюванні параметрів блоку.

Якщо операнди є скалярними величинами, то на виході блоку формується скалярний сигнал, значення якого визначається результатом операції - 1, якщо результат операції «істина» (TRUE), і 0, якщо результат операції «неправда» (FALSE).

Якщо операнди є векторними величинами, то операція виконується поелементно; у зв'язку із цим векторні операнди повинні мати однакову розмірність. Результат операції над векторами є вектором тієї ж розмірності, але складається тільки з булевих величин (0 й 1).

5.14. Logical Operator виконує логічну операцію (AND, OR, NAND, NOR, XOR або NOT) із вхідними булевими величинами.

Блок містить набір основних логічних операторів: AND (логічне множення I); OR (логічне додавання АБО); NAND(I-НЕ); NOR(АБО-НІ); XOR (додавання по модулі 2); NOT (логічне заперечення).

Параметр Number of input ports (число вхідних портів блоку) задає число вхідних величин, тобто число аргументів логічної операції.

Якщо операнди є скалярними величинами, то на виході блоку формується скалярний сигнал, значення якого визначається результатом операції - 1, якщо результат операції «істина» (TRUE), і 0, якщо результат операції «неправда» (FALSE).

Якщо операнди є векторними величинами, то операція виконується поелементно; у зв'язку із цим векторні операнди повинні мати однакову розмірність. Результат операції над векторами є вектором тієї ж розмірності, але складається тільки з булевих величин (0 та 1).

5.15. Combinatorial Logic (комбінаторика) забезпечує перетворення вхідних булевих величин у вихідну відповідно до таблиці істинності, яка повинна бути задана користувачем.

Вхідні дані й результат повинні бути булевими величинами. На вхід може подаватися скалярний або векторний сигнал. Будь-яке ненульове значення вхідного сигналу відповідає значенню «істина».

Параметр налаштування, Truth table (таблиця істинності), це список можливих вихідних значень.

Наприклад, якщо на вхід блоку Combinatorial Logic подається векторний двомірний сигнал, то Truth table буде складатися з чотирьох елементів, наприклад таких: [0; 1; 0; 1] або можливо таких: [11, 01, 10, 11] тощо.

Можливі таблиці істинності Combinatorial Logic із двомірним вектором на вході:

Вхідний сигнал		Можливий вихідний сигнал	Можливий вихідний сигнал
1-й елемент	2-й елемент		
0	0	0	11
0	1	1	01
1	0	0	10
1	1	1	11

Таким чином, параметр Truth table описує значення тільки вихідного сигналу. Розрядність вихідного сигналу, а також значення кожного його розряду вибираються користувачем на основі власних міркувань про логіку роботи створюваної моделі.

Блоки ланок з типовими нелінійностями.

5.16. Sign реалізує нелінійність типу сигнум-функції.

Формує вихідний сигнал, що приймає тільки три можливих значення: +1 - у випадку, коли вхідний сигнал позитивний, -1 - при негативному вхідному сигналі й 0 - при вхідному сигналі рівному нулю.

5.17. Gain перемножає вхідний сигнал на заданий користувачем коефіцієнт.

Параметри настроювання блоку:

- Gain числова константа, змінна або вираз, який обчислюється.
- Saturate on integer overflow опрацьовувати переповнення для цілих, якщо результат перевищує значення припустиме для цілих.

5.18. Slider Gain (повзунковий регулятор) — ланка інтерактивної зміни коефіцієнта підсилення.

Має три поля введення, які задають: нижня межа зміни параметра (Low); верхня межа зміни параметра (High); поточне значення. Поточне значення повинно лежати усередині діапазону. При виборі нового діапазону необхідно спочатку вказати нове значення параметра, а потім змінити границі діапазону.

5.19. Matrix Gain (матричний перемножувач) множить вхідний сигнал на матрицю (або вектор), ім'я якої зазначене як значення параметра блоку — Gain matrix (матриця, яка буде помножена).

Блок реалізує закон $\bar{y} = K \cdot \bar{x}$, де x, y – вектори, K – матриця.

5.20. Algebraic Constraint (алгебраїчний корінь) для відшукування коренів алгебраїчних рівнянь. Спільне використання декількох блоків забезпечує рішення системи алгебраїчних рівнянь.

Блок обнуляє вхідний сигнал шляхом одержання такого вихідного сигналу, що, будучи поданим на вхід системи, приводить до обнуління входу блоку Algebraic Constraint. Для нормальної роботи блоку система

повинна мати зворотний зв'язок, щоб вихід блоку міг впливати на його вхід.

Блок можна використати для рішення лінійних і нелінійних алгебраїчних рівнянь і систем. Пошук кореня носить ітераційний характер. Початкове значення задається параметром Initial guess (початкове наближення).

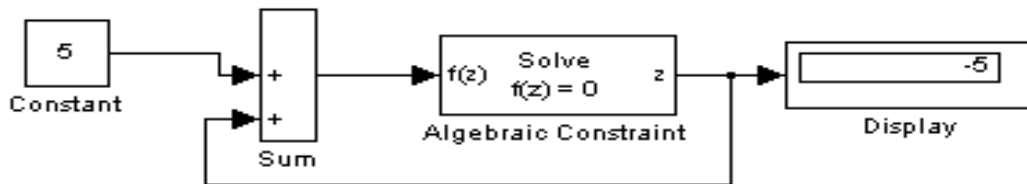


Рис. 4. Блок-модель для рішення рівняння $z+5=0$

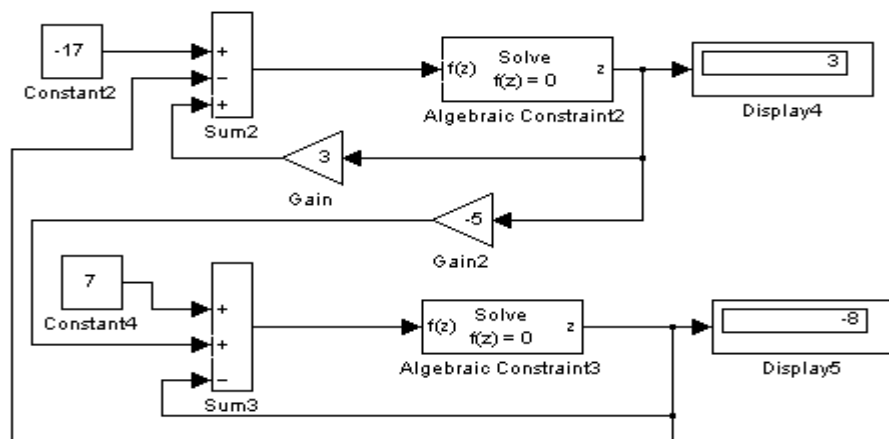


Рис. 5. Рішення системи алгебраїчних рівнянь

$$\begin{cases} 3z_1 - z_2 - 17 = 0 \\ -5z_1 + z_2 + 7 = 0 \end{cases}$$

6. Розділ Discrete (дискретні системи)

Блоки для моделювання дискретних систем і перетворення безперервних систем у дискретні:

6.1. Unit Delay — блок затримки сигналу на один період дискретизації.

Параметри блоку: Initial condition (Початкове значення сигналу) і Sample time (Час затримки), що задає числом кроків модельного часу.

6.2. Discrete-Time Integrator — дискретний інтегратор.

Виконує підсумовування інтервалів часу між надходженнями вхідного сигналу. Може бути використаний для керування логікою роботи окремих компонентів або моделі в цілому. Зокрема, сумарна тривалість роботи деякої підсистеми може служити умовою «дострокового» закінчення моделювання.

Більшість параметрів налаштування цього блоку збігаються з параметрами блоку Integrator розділу Linear. Відмінності полягають у наступному. У блоці дискретного інтегратора є додатковий параметр - метод чисельного інтегрування (Integrator method). За допомогою списку можна вибрати один із трьох методів:

- прямий метод Ейлера (лівих прямокутників);
- зворотний метод Ейлера (правих прямокутників);
- метод трапецій.

Друга відмінність - замість параметра Absolute tolerance уведений параметр Sample time, що задає крок інтегрування в одиницях кроків модельного часу.

6.3. Zero-Order Hold — екстраполятор нульового порядку.

6.4. First-Order Hold — екстраполятор (фіксатор) першого порядку.

6.5. Discrete State-Space — блок задання дискретної ланки матрицями її стану.

Дискретна система, описана в просторі станів:

$$\begin{cases} x(n+1) = Ax(n) + Bu(n) \\ y(n) = Cx(n) + Du(n) \end{cases}$$

Параметри настроювання блоку:

- A, B, C, D - значення матриць A, B, C, D;
- Initial Condition - початковий стан;
- Sample time - період дискретизації.

6.6. Discrete Filter — блок задання дискретної ланки через дискретну передатну раціональну-дрібно-раціональну функцію відносно $1/z$.

Блок реалізує цифровий фільтр. Передатна функція має вигляд:

$$W(z) = \frac{k_1 + k_2 z^{-1} + k_3 z^{-2} + \dots}{m_1 + m_2 z^{-1} + m_3 z^{-2} + \dots}.$$

Параметри настроювання блоку:

- Numerator (коефіцієнти чисельника;
- Denominator (коефіцієнти знаменника;
- Sample time (період дискретизації.

6.7. Discrete Transfer Fcn — блок завдання дискретної ланки через дискретну передатну дрібно-раціональну функцію відносно z .

Дискретна передатна функція має вид $W(z) = \frac{P(z)}{Q(z)}$.

Параметри настроювання блоку:

- Numerator - коефіцієнти чисельника;
- Denominator - коефіцієнти знаменника;
- Sample time - період дискретизації

6.8. Discrete Zero-Pole — блок завдання дискретної ланки через значення нулів і полюсів дискретної передатної функції відносно $1/z$.

Параметри настроювання блоку:

- Zeros - Вектор нулів;
- Poles - вектор полюсів;
- Gain - коефіцієнт передачі;

- Sample time - період дискретизації

7. Розділ Signals&Systems (сигнали й системи)

Блоки, що забезпечують керування сигналами:

7.1. Bus Selector (селектор шини) — виділяє із групи сигналів один заданий сигнал.

У вікні налаштування блоку є два списки - вхідний і вихідний.

7.2. Mix (змішувач) поєднує вхідні сигнали в один вихідний вектор. Параметр налаштування - Number of inputs (число входів).

7.3. Demux (роздільник) розділяє вхідний вектор на задане число компонентів. Параметр налаштування Number of outputs (Число виходів).

7.4. Matrix Concatenation — векторний сигнал, представлений розгорнутим рядком або стовпцем, перетворює до «згорнутого» векторному сигналу.

7.5. Reshape — змінює розмірність вхідного сигналу.

7.6. Function-Call Generator — забезпечує запуск підключених до нього підсистем із заданою періодичністю.

7.7. Configurable Subsystem — реалізує функцію будь-якої підсистеми, що може бути бібліотечною або створеною користувачем.

7.8. Signal Specification керує процесом моделювання залежно від параметрів вхідного сигналу.

7.9. Model Info — дозволяє одержати інформацію про моделі.

7.10. From (прийняти), **Goto Tag Visibility** (ознака видимості), **Goto** (передати) використовуються спільно й призначені для обміну даними між різними частинами моделі з урахуванням доступності цих даних.

Блок From - приймач сигналу, переданого блоком Goto. Застосовується для розриву ліній зв'язку. Параметр Goto tag - мітка переходу (ім'я сигналу).

7.11. Goto (передати) - передавач сигналу. Приймачем сигналу є блок From. Застосовується для розриву ліній зв'язку.

Параметри налаштування блоку:

- Tag - мітка переходу (ім'я сигналу);
- Tag visibility - видимість мітки:
 - local - блоки передавача й приймача повинні перебувати в одній підсистемі. Мітка переходу в зображенні такого блоку береться у квадратні дужки.
 - scored - блоки передавача й приймача повинні перебувати в одній підсистемі або в будь-якій підсистемі нижче по ієрархії моделей від підсистеми, що містить блок Goto Tag Visibility. Мітка переходу в зображенні такого блоку береться у фігурні дужки.
 - global - блоки передавача й приймача можуть перебувати в будь-якій підсистемі системи

7.12. Goto Tag Visibility (ознака видимості) задає видимість мітки переходу.

Блоки передавача з параметром Tag Visibility, рівним scored і приймача повинні перебувати в одній підсистемі або в будь-якій

підсистемі нижче по ієрархії моделей від підсистеми, що містить блок Goto Tag Visibility. Параметр Goto Tag - мітка переходу (ім'я сигналу).

7.13. Data Store Read (читання даних), **Data Store Memory** (запам'ятовування даних) і **Data Store Write** (Запис даних) використовуються спільно й забезпечують передачу даних та їхнє зберігання на інтервалі моделювання.

Блоки для організації спільної області пам'яті для обміну між підсистемами однієї системи. У кожній з підсистем, що використовують блоки Data Store Read й Data Store Write повинен бути визначений блок Data Store Memory.

Параметри налаштування блоку:

- Data store name - ім'я блоку пам'яті
- Initial value - початкове значення
- Sample time - період дискретизації

7.14. IC (Initial Condition — початкова умова) установлює початкове значення сигналу, причому тривалість перебування системи у відповідному стані дорівнює тривалості кроку моделювання, а потім блок пропускає вхідний сигнал на вихід без зміни.

7.15. Selector (селектор) вибирає у вхідному векторі й передає на вихід тільки ті елементи, номери яких зазначені в параметрах налаштування блоку.

Істотним достоїнством є те, що значення параметрів його налаштування відображаються в графічній формі на зображенні блоку.

Параметри налаштування блоку:

- Elements - вектор, що вказує на порядок елементів вихідного вектора;
- Input port width - ширина вхідного вектора.

7.16. Width (Розмір) визначає розмірність сигналу, що надходить на його вхід. Значення розмірності виводиться безпосередньо на зображенні блоку.

Вихідний сигнал блоку рівний кількості елементів (ширині) вхідного вектора.

7.17. Merge (злиття) виконує об'єднання вхідних сигналів в один.

Блок комбінує вхідні сигнали в скалярний вихідний сигнал. Параметр Number of inputs - кількість вхідних портів.

7.18. Hit Crossing (Виявити перетинання) дозволяє зафіксувати стан, коли вхідний сигнал перетинає деяке значення.

При виникненні такої ситуації на виході блоку формується одиничний сигнал.

Параметри настроювання блоку:

- Hit crossing offset (поріг перетинання) - визначає числове значення, перетинання якого необхідно ідентифікувати;
- Hit crossing direction - дозволяє вказати форму зміни вхідного сигналу, при якому це перетинання повинне виявлятися: rising (зростання), falling (убування), either (у будь-якому напрямку);
- Show output port (показати вихідний порт) — опція, за допомогою якої вибирається формат використання блоку.

При одночасному виконанні умов, що задають параметрами Hit crossing offset й Hit crossing direction, на виході блоку формується одиничний імпульс. Його тривалість визначається значенням параметра Sample time (еталонний час) блоку, що передує в моделі блоку Hit crossing. Якщо цей параметр відсутній, то одиничний сигнал на виході блоку Hit crossing залишається до його наступного спрацьовування.

8. Розділ Functions & Tables (сигнали й системи)

8.1. Fcn дозволяє ввести функцію від одного (скалярного або векторного) аргументу виразу через стандартні функції MatLAB.

Вираз функції вводиться у вікні налаштування блоку. Вхідний сигнал (аргумент функції) позначається символом *u*.

У якості операндів можуть використовуватись змінні робочої області MATLAB. Наприклад, *u(1)+ans*. Стиль запису виразів подібний зі стилем мови програмування C.

Можливо використання операторів відношення (*=*, *!=*, *>*, *<*, *>=*, *<=*) і логічних операторів (*&&*, *|*, *!*), результат буде 1 або 0.

Значення вираз, що обчислюється, повинно бути скалярним. Це обмеження знімається при використанні блоку MATLAB Fcn.

8.2. MATLAB Fcn дозволяє застосувати до вхідного сигналу підпрограму обробки, реалізовану у вигляді М-файлу.

Параметри налаштування блоку:

- MATLAB function - ім'я М-файлу або вираз, для обчислення.
- Output width (тривалість вихідного сигналу). Якщо тривалість вихідного сигналу збігається із тривалістю вхідного, то значення даного параметра встановлюється рівним -1. Якщо це не так, то необхідно заздалегідь визначити тривалість вихідного сигналу й указати її як значення параметра. При невідповідності розмірності сигналу встановленому значенню SIMULINK видає повідомлення про помилку.
- Output signal type (числовий формат подання амплітуди вихідного сигналу: *real* (сигнал інтерпретується як дійсне число); *complex* (комплексне); *auto*).

8.3. Look-up Table – завдання функції у вигляді одновимірної таблиці.

Значення, що не попадають у задані точки, інтерполюються.
Значення, що перебувають поза областю заданих точок, екстраполуються.

Параметри налаштування блоку:

- Vector of input values - вектор вхідних величин;
- Vector of output values - вектор вихідних величин.

8.4. Look-up Table(2D) – завдання функції у вигляді двовірної таблиці.

Значення, що не попадають у задані точки, інтерполюються.
Значення, що перебувають поза областю заданих точок, екстраполуються.

Параметри налаштування блоку:

- Row - рядки;
- Column - стовпці;
- Table - таблиця значень.

8.5. Look-Up Table (n), PreLook-Up Index Starch, Interpolation (n) using PreLook-Up, Direct Look-Up Table (n) — блоки, що виконують різного виду інтерполяції.

8.6. S-Function — блок, що дозволяє створювати ієрархічну структуру для складних моделей, поєднуючи групу блоків в один.

У даний блок включаються власні програми, написані мовою C, які зв'язуються з бібліотечними блоками пакета Simulink;

8.7. Polynomial — багаточлен ступеня n, заданий своїми коефіцієнтами. У якості незалежної змінної в багаточлені виступає вхідна величина.

9. Розділ Subsystems (сигнали й системи)

Блоки для побудови складних моделей, що складаються з підсистем

9.1. Enable (дозволити) для створення підсистем з дозволом. Підсистема з дозволом функціонує тільки тоді, коли значення сигналу дозволу більше нуля.

Параметри налаштування блоку:

- States when enabled - початковий стан підсистеми в момент переходу із забороненого стану в дозволене:
 - held - стан підсистеми в попередній інтервал часу, коли підсистема була дозволена;
 - reset - скидання стану підсистеми;
- Show output port - показати вихідний порт (він індикуює стан сигналу дозволу)

9.2. Trigger (засувка) призначений для логічного керування роботою підсистем моделі

Блок призначений для створення тригерних (синхронізуємих) підсистем. Така підсистема виконується тільки на тих кроках інтегрування, на яких сигнал на тригерному вході підсистеми змінюється, як задано в параметрі Trigger Type. Підсистема не може містити більше одного блоку Trigger.

Параметри налаштування блоку:

- Trigger type - тип тригера (тип керуючого сигналу):
 - rising - підсистема виконується, якщо керуючий сигнал міняється з негативного або нульового до позитивного або з негативного до нульового;
 - falling - підсистема виконується, якщо керуючий сигнал міняється з позитивного або нульового до негативного або з позитивного до нульового;

- either - підсистема виконується при обох умовах (rising й falling);
- function call - виконання підсистеми, керованою S-функцією;
- Show output port - показати вихідний порт;
- Output data type - тип вихідних даних:
 - auto - тип визначається автоматично;
 - double - подвійна точність;
 - int8 - 8-бітне ціле.

9.3. Subsystem (підсистема) являє собою заготовку для створення підсистеми.

Дозволяє представити кілька блоків, об'єднаних в одній бібліотеці.
Параметр Library name - ім'я бібліотеки.

9.4. Discrete Pulse Generator— дискретний генератор послідовності прямокутних імпульсів.

Параметри налаштування блоку:

- Amplitude - амплітуда сигналу (висота прямокутного імпульсу);
- Period - період, відлічуваний у цілих числах (у дискретах часу);
- Pulse width - ширина імпульсу (також у дискретах часу);
- Phase delay - величина затримки імпульсу відносно $t=0$ (у дискретах часу);
- Sample time - величина дискрети часу.

Установку параметрів блоку доцільно починати з установки дискрета часу.

**Блоки SIMULINK, розсортовані за алфавітом,
(за номером блоку можна знайти детальніший його опис)**

№	Блок	Розділ
5.1.	Abs формує абсолютне значення вхідного сигналу.	Math
5.20.	Algebraic Constraint (алгебраїчний корінь) для відшукування коренів алгебраїчних рівнянь. Спільне використання декількох блоків забезпечує рішення системи алгебраїчних рівнянь.	Math
3.2.	Backlash (люфт) – забезпечує «утримання» вхідного сигналу в деякому заданому діапазоні.	Nonlinear
1.1.	Band-Limited White Noise – джерело білого частотно-обмеженого шуму.	Sources
7.1.	Bus Selector (селектор шини) — виділяє із групи сигналів один заданий сигнал.	Signals & Systems
1.2.	Chirp Signal – джерело синусоїдального сигналу одиничної амплітуди й змінної частоти, частота змінюється в часі за лінійним законом.	Sources
1.4.	Clock — генератор безперервного сигналу часу. Його робота заснована на використанні апаратного таймера комп'ютера	Sources
5.15.	Combinatorial Logic (комбінаторика) забезпечує перетворення вхідних булевих величин у вихідну відповідно до таблиці істинності заданої користувачем.	Math
5.6.	Complex to Magnitude-Angle (виділення з комплексного числа амплітуди й фази) дозволяє одержати із вхідної комплексної величини амплітудно-фазові характеристики сигналу.	Math
5.8.	Complex to Real-Image (виділення мнімої або дійсної	Math

	частини комплексного числа) забезпечує виділення із вхідного комплексного сигналу його мнімої або дійсної частини (залежно від настроювання параметрів блоку).	
7.7.	Configurable Subsystem — реалізує функцію будь-якої підсистеми, що може бути бібліотечною або створеною користувачем.	Signals & Systems
1.3.	Constant — джерело постійної величини (скаляра, вектора або матриці).	Sources
3.10.	Coulomb & Viscous Friction (Сухе й вязке тертя) реалізує нелінійну залежність типу лінійна з попереднім натягом.	Nonlinear
7.13.	Data Store Read (читання даних), Data Store Memory (запам'ятовування даних) і Data Store Write (Запис даних) використовуються спільно й забезпечують передачу даних й їхнє зберігання на інтервалі моделювання.	Signals & Systems
3.1.	Dead Zone (пасивна зона) заміняє значення вхідного сигналу лежаче в заданому діапазоні, нулем, а значення лежаче в заданому діапазоні наближає до границь цього діапазону, змінюючи їх.	Nonlinear
7.3.	Demux (роздільник) розділяє вхідний вектор на задане число компонентів. Параметр настроювання Number of outputs (Число виходів).	Signals & Systems
4.4.	Derivative (похідна) — блок обчислення похідної вхідного сигналу за часом (du/dt).	Continuous
1.5.	Digital clock — джерело дискретного сигналу часу, тобто час моделі буде змінюватися тільки в задані моменти часу.	Sources
6.6.	Discrete Filter — блок завдання дискретної ланки через дискретну передатну дрібно-раціональну функцію відносно	Discrete

	1/z.	
9.4.	Discrete Pulse Generator — дискретний генератор послідовності прямокутних імпульсів.	Subsystems
6.5.	Discrete State-Space — блок завдання дискретної ланки матрицями її стану.	Discrete
6.7.	Discrete Transfer Fcn — блок завдання дискретної ланки через дискретну передатну дрібно-раціональну функцію відносно z.	Discrete
6.8.	Discrete Zero-Pole — блок завдання дискретної ланки через вказівку значень нулів і полюсів дискретної передатної функції відносно 1/z.	Discrete
6.2.	Discrete-Time Integrator — дискретний інтегратор.	Discrete
2.3.	Display (екран) – відображення чисельних значень величин.	Sinks
5.12.	Dot product (згортка) – скалярне множення двох векторів, наприклад, згортка вектора [2 3] й [1 4] дорівнює 14.	Math
9.1.	Enable (дозволити) для створення підсистем з дозволом. Підсистема з дозволом функціонує тільки тоді, коли значення сигналу дозволу більше нуля.	Subsystems
8.1.	Fcn дозволяє ввести функцію від одного (скалярного або векторного) аргументу, яка виражається через стандартні функції MatLAB.	Functions & Tables
6.4.	First-Order Hold — екстраполятор (фіксатор) першого порядку.	Discrete
7.10.	From (прийняти), Goto Tag Visibility (ознака видимості), Goto (передати) використовуються спільно й призначені для обміну даними між різними частинами моделі з урахуванням доступності цих даних.	Signals & Systems

1.6.	From File — уведення в блок-модель даних, що зберігаються в mat-файлі.	Sources
1.7.	From Workspace — уведення в модель даних з робочої області MATLAB.	Sources
7.6.	Function-Call Generator — забезпечує запуск підключених до нього підсистем із заданою періодичністю.	Signals & Systems
5.17.	Gain перемножає вхідний сигнал на коефіцієнт, заданий користувачем.	Math
7.11.	Goto (передати) - передавач сигналу. Приймачем сигналу є блок From. Застосовується для розриву ліній зв'язку.	Signals & Systems
7.12.	Goto Tag Visibility (ознака видимості) задає видимість мітки переходу.	Signals & Systems
1.8.	Ground (земля) і Terminator (обмежник) можуть використовуватись як заглушки для тих портів, які з якої-небудь причини виявилися не підключеними до інших блоків моделі. Причому Ground використовується для вхідних портів, а Terminator — для вихідних портів.	Sources
7.18.	Hit Crossing (Виявити перетинання) дозволяє зафіксувати стан, коли вхідний сигнал перетинає деяке значення.	Signals & Systems
1.9.	In (вхідний порт) вхідний порт підсистеми, разом з Out (вихідний порт) забезпечують зв'язок між підсистемами моделі за інформацією.	Sources
4.8.	Integrator — ідеальна інтегруюча ланка, здійснює інтегрування в безперервному часі вхідної величини	Continuous
7.14.	IC (Initial Condition — початкова умова) задає початкове значення сигналу, причому тривалість перебування системи у відповідному стані дорівнює тривалості кроку	Signals & Systems

	моделювання, а потім блок пропускає вхідний сигнал на вихід без зміни.	
5.14.	Logical Operator виконує логічну операцію (AND, OR, NAND, NOR, XOR або NOT) із вхідними булевими величинами.	Math
8.5.	Look-Up Table (n), PreLook-Up Index Starch, Interpolation (n) using PreLook-Up, Direct Look-Up Table (n) — блоки, що виконують різного виду інтерполяції.	Functions & Tables
8.3.	Look-up Table – задання функції у вигляді одновірної таблиці.	Functions & Tables
8.4.	Look-up Table(2D) – задання функції у вигляді двовірної таблиці.	Functions & Tables
5.7.	Magnitude-Angle to Complex (перетворення амплітуди й фази в комплексне число).	Math
3.6.	Manual Switch – ручний перемикач.	Nonlinear
5.3.	Math Function задає для перетворення сигналу елементарні не тригонометричні й не гіперболічні функції, такі як обчислення експоненти, натурального й десяткового логарифмів, піднесення в ступінь, добування квадратного кореня й т.д.	Math
8.2.	MATLAB Fcn дозволяє застосувати до вхідного сигналу підпрограму обробки, реалізовану у вигляді М-файлу.	Functions & Tables
7.4.	Matrix Concatenation — векторний сигнал, представлений розгорнутим рядком або стовпцем, перетворює до «згорнутого» векторному сигналу.	Signals & Systems
5.19.	Matrix Gain (матричний перемножувач) перемножує вхідний сигнал на матрицю (або вектор), ім'я якої зазначене	Math

	як значення параметра блоку — Gain matrix (матриця, яка перемножається).	
4.5.	Memory (регістр) — затримка сигналу на один крок модельного часу.	Continuous
7.17.	Merge (злиття) виконує об'єднання вхідних сигналів в один.	Signals & Systems
5.5.	MinMax здійснює пошук мінімального або максимального елемента вхідного вектора.	Math
7.9.	Model Info — дозволяє одержати інформацію про моделі.	Signals & Systems
3.7.	Multiport Switch – багатовходовий перемикач.	Nonlinear
2.8.	Out (вихідний порт) – вихідний порт підсистеми, разом з In (вхідний порт) забезпечують зв'язок між підсистемами моделі за інформацією.	Sinks
8.7.	Polynomial — багаточлен ступеня n , заданий своїми коефіцієнтами. У якості незалежної змінної в багаточлені виступає вхідна величина.	Functions & Tables
5.10.	Product виконує множення або ділення декількох вхідних сигналів.	Math
1.10.	Pulse Generator — генератор імпульсних сигналів прямокутної форми.	Sources
3.9.	Quantizer (Квантувач) здійснює дискретизацію вхідного сигналу по його величині.	Nonlinear
1.11.	Ramp — генератор лінійно зростаючого (убуваючого) сигналу.	Sources
1.12.	Random Number — джерело дискретного сигналу, амплітуда якого є випадковою величиною, розподіленою за	Sources

	нормальним законом.	
3.3.	Rate Limiter (обмежник швидкості) – для контролю й обмеження швидкості зміни (тобто першої похідної) вхідного сигналу.	Nonlinear
5.9.	Real-Image to Complex (створення комплексного числа з його мнімої й дійсної частини).	Math
5.13.	Relational Operator реалізує операції відношення над двома вхідними сигналами.	Math
3.8.	Relay працює аналогічно реле: якщо вхідний сигнал перевищує граничне значення, то на виході формується «дозволяючий» сигнал.	Nonlinear
1.13.	Repeating Sequence — генератор періодичного дискретного сигналу, форма якого задається в параметрах блоку і являє собою ламану лінію, що проходить через точки із зазначеними у векторах Time values й Output values координатами.	Sources
7.5.	Reshape — змінює розмірність вхідного сигналу.	Signals & Systems
5.4.	Rounding Function — виконує різні функції округлення.	Math
3.4.	Saturation (насичення) — вихідна величина цього блоку збігається із вхідною, якщо остання перебуває усередині зазначеного діапазону. Якщо ж вхідна величина виходить за рамки діапазону, то вихідний сигнал приймає значення найближчої із границь. Значення границь діапазону встановлюються у вікні налаштування блоку.	Nonlinear
2.1.	Scope – осцилограф (індикатор).	Sinks
7.15.	Selector (селектор) вибирає у вхідному векторі й передає на	Signals &

	вихід тільки ті елементи, номери яких зазначені в параметрах налаштування блоку.	Systems
8.6.	S-Function — блок, що дозволяє створювати ієрархічну структуру для складних моделей, поєднуючи групу блоків в один.	Functions & Tables
5.16.	Sign реалізує нелінійність типу сигнум-функції.	Math
1.14.	Signal Generator — генератор безперервного сигналу довільної форми.	Sources
7.8.	Signal Specification — управляє процесом моделювання залежно від параметрів вхідного сигналу.	Signals & Systems
1.16.	Sine Wave — генератор гармонійних коливань.	Sources
5.18.	Slider Gain (повзунковий регулятор) — ланка інтерактивної зміни коефіцієнта підсилення.	Math
4.2.	State-Space (простір станів) — блок формування стану системи.	Continuous
1.15.	Step — генератор типового впливу у вигляді одиничної сходинок.	Sources
2.6.	Stop Simulation – зупинка моделювання, при надходженні на його вхід ненульового сигналу. Не має параметрів налаштування.	Sinks
9.3.	Subsystem (підсистема) являє собою заготовку для створення підсистеми.	Subsystems
5.11.	Sum — ланка-суматор.	Math
3.5.	Switch – перемикає до виходу один із двох інформаційних входів залежно від сигналу на керуючому вході.	Nonlinear
2.7.	Terminator (обмежник) і Ground (земля) можуть використовуватись як заглушки для тих портів, які з якої-	Sinks

	небудь причини виявилися не підключеними до інших блоків моделі. Причому Ground використовується для вхідних портів, а Terminator — для вихідних портів.	
4.1.	Transfer Fcn — реалізує передатну функцію виду $W(s)=P(s)/Q(s)$.	Continuous
4.6.	Transport Delay (затримка передачі) — забезпечує затримку сигналу на задану кількість кроків модельного часу, причому не обов'язково ціле.	Continuous
9.2.	Trigger (засувка) призначений для логічного керування роботою підсистем моделі	Subsystems
5.2.	Trigonometric забезпечує перетворення вхідного сигналу за допомогою однієї з функцій: sin, cos, tan, asin, acos, atan, sinh, cosh, tanh.	Math
1.17.	Uniform Random Number — джерело дискретного сигналу, амплітуда якого є рівномірно розподіленою випадковою величиною.	Sources
6.1.	Unit Delay — блок затримки сигналу на один період дискретизації.	Discrete
4.7.	Variable Transport Delay (плаваюча затримка передачі) — блок, що реалізує динамічно змінювану затримку вхідного сигналу.	Continuous
7.16.	Width (Розмір) визначає розмірність сигналу, що надходить на його вхід. Значення розмірності виводиться безпосередньо на зображенні блоку.	Signals & Systems
2.2.	XYGraph (двовимірний графік) – створення двовимірних графіків залежностей довільних величин, що фігурують у моделі, у прямокутній системі координат.	Sinks

6.3.	Zero-Order Hold — екстраполятор нульового порядку.	Discrete
4.3.	Zero-Pole (область переходу) — блок, що реалізує передатну функцію, описану за допомогою оператора Лапласа у вигляді полюсів, нулів і коефіцієнта передачі.	Continuous
7.2.	Mux (змішувач) поєднує вхідні сигнали в один вихідний вектор. Параметр налаштування — Number of inputs (число входів).	Signals & Systems
2.4.	To File — забезпечує запис значень величини, поданої на його вхід, в mat-файл.	Sinks
2.5.	To Workspace — запис даних у робочу область MatLAB.	Sinks

Список літератури

1. Гультяев А. Визуальное моделирование в среде MATLAB: учебный курс. — СПб: Питер, 2000. — 432 с.
2. Лазарев Ю.Ф. MatLAB 5.x. — К.: Издательская группа BHV, 2000. — 384 с.
3. Лазарев Ю.Ф. Моделирование процессов и систем в MatLAB. Учебный курс. — СПб.: Питер; К.: Издательская группа BHV, 2005. — 512 с.
4. Краснопрошина А.А., Репникова Н.Б., Ильченко А.А. Современный анализ систем управления с применением MATLAB, Simulink, Control System: Учебное пособие. — К.: "Корнійчук", 1999. — 144 с.
5. Топчеев Ю.И. Атлас для проектирования систем автоматического регулирования: Учебное пособие для втузов. — М.: Машиностроение, 1989. — 759 с.
6. Герман-Галкин С.Г. Компьютерное моделирование полупроводниковых систем в MATLAB 6.0: Учебное пособие — СПб.: КОРОНА принт, 2001. — 320 с.

7. Герман-Галкин С. Г., Кардонов Г. А. Электрические машины: Лабораторные работы на ПК. – СПб.: КОРОНА принт, 2003. – 256 с.